**DOCTEUR EN SCIENCES**

# SOA Method Engineering Framework for process-oriented and model-driven SOA implementation with underlying models of abstraction

Ricken, Jan

*Award date:*
2013

*Awarding institution:*
Universite de Namur

## University of Namur, Computer Science Department

---



# PhD Thesis on SOA Methods for Process-Oriented Implementation

> ***"SOA Method Engineering Framework for process-oriented and model-driven SOA implementation with underlying models of abstraction"***

**PhD Thesis prepared by Jan Ricken for achieving the grade of "Docteur en Science Informatique".**

**Date of Public Defense : 21/05/2013**

**PReCISE Research Center**
**Computer Science Department, University of Namur**

**PhD Thesis Jury:**

**Prof. Michael Petit (Supervisor)**
**Prof. Eric Dubois (Co-Supervisor)**
**Prof. Naji Habra (DEAN of Computer Science Department)**
**Prof. Vincent Englebert (Computer Science Department)**
**Prof. Philippe Thiran (Computer Science Department)**
**Prof. Henderik A. Proper (External Evaluator)**

# Table of Contents

# Table of Figures

# Table of Tables

# Table of Definitions

# Table of Abbreviations

AD: Architecture Description
AF: Architecture Framework
ARIS: Architecture for Integrated Systems
ATHENA: Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications
AVE: Aris Value Engineering
BMM: Business Motivation Model
BPDM: Business Process Definition Metamodel
BPEL: Business Process Execution Language for Web Services
BPM: Business Process Management
BPMI: Business Process management Initiative
BPMN: Business Process Modeling Notation
BSC: Balanced Scorecard
CAM-I: Computer Aided Manufacturing – International
CIM: Computation Independent Model
CIMOSA: Computer Aided Manufacturing – Open System Architecture
CIO: Chief Information Officer
CORBA: Common Object Request Broker Architecture
CSF: Critical Success Factors
CWM: Common Warehouse Meta-Model
DCOM: Distributed Component Object Model
EA: Enterprise Architecture
EAI: Enterprise Applications Integration
EM: Enterprise Modelling
EML: Enterprise Modelling Languages
EMM: Enterprise Modelling Methods
EMT: Enterprise Modelling Tools
EPC: Event-Driven-Process Chains
ESB: Enterprise Service Bus
FTE: Full Time Equivalent
ICAM: Integrated Computer Aided Manufacturing
IDEF: Integrated Computer-Aided Manufacturing Definition – family
IEEE: Institute of Electrical and Electronics Engineers
INTEROP: Interoperability of Enterprise Software and Applications
KPI: Key Performance Indicators
MDA: Model Driven Architecture
MDM: Master Data Management
MDSD: Model-Driven Software Development
ME: Method Engineering
MOF: Meta Object Facility
OMG: Object Management Group
PIM: Platform-Independent Model
PIM4SOA: Platform-independent model for service-oriented architecture
POSA: Patterns of Software Architecture
PSL: Process Specification Language
PSM: Platform Specific Model
QoS: Quality of Web-Services
REST: Representational State Transfer

ROI: Return of Investment
RUP: Rational Unified Process
SME: Situational Method Engineering
SLA: Service Level Agreement
SOA: Service Oriented Architecture
SoDD: Service-oriented Design and Development Methodology
SOMA: Service –Oriented Modeling and Architecture
SPEM: Software Process Engineering Metamodel
TD-MD-PO: Top-Down, Model-Driven and Process-Oriented
UEML: Unified Enterprise Modelling Language
UML: Unified Modeling Language
VACD: Value-added chain diagram
VbMF: view-based modeling framework
WOA: Web-Oriented Architecture
WS: Web Services
WSDL Web Services Description Language
XPDL: XML Process Definition Language
YAWL: Yet Another Workflow Language

# CHAPTER 1

# INTRODUCTION



This chapter introduces the thesis by explaining the motivation for the chosen research topic (section 1.1.) with the background and proposed research, why SOA is important and where the problems are. Then, the proposed research contribution is explained (section 1.2.) including research objectives and research questions and how this thesis will contribute to resolve the identified research problem. The scope of work is detailing the research design (section 1.3.) and the overall research structure to ensure efficient organization and usage of research standards.

## *1.1. Motivation*

### 1.1.1. Background of Research

Recently, we can observe that established business rules have been constantly redefined [KBS06] and organizations are constantly hunting for effectiveness and efficiency in their daily operation. New business models emerged: speed, growth and innovation were the critical success factors to survive the wave of mergers & acquisitions that changed the overall industrial landscape. IT has played a major role in all of this and without any doubt IT is the key enabler for the hot topic of business agility. The agility and flexibility [Sch04] of an organization today to react as quickly as possible on different opportunities & threats (mergers & acquisitions) strategic partnership and alliances, new product launches which are based on customer requirements is key. This agility and flexibility turned into a success factor to survive in the aggressive and competitive markets pushed by the globalisation and the European Union enlargement. In this context, Service Oriented Architecture (SOA) became since the early 1990s an interesting philosophy [Pez06].

Initially pioneered on technologies such as peer-to-peer network protocols, SOA got a boost in adoption in the second half of the 1990s. Enabled by Common Object Request Broker Architecture (CORBA) and Distributed Component Object Model (DCOM), SOA began to become a more popular concept and reached another step in maturity with the development of platforms such as Java 2 Platform, Enterprise Edition (J2EE) and .NET, increased adoption of Business Process Management (BPM) and the emergence of Web Services (WS). The SOA paradigm is defined as an

**Definition 1: SOA [Cha03] [Bar03]**

> *"architectural concept in which all functions, or services, are defined using a description language and have invokable, platform-independent interfaces that are called to perform business processes [Cha03] [Bar03]".*

Through this, the concept of SOA became more mature and reached the top of the hype cycle in 2003 [Pez06]. The SOA paradigm has been identified as the magic bullet to first and foremost enable rapidly an adaptation to the quickly changing business environment [Alo04]. SOA meets the definition of an architectural style and represents a technical view of a business automation solution based on service-orientation principles [Erl05]. An architectural style needs to be understood as "a group of principles that provides a framework for a family of systems" [Pon12]. The key principles of service orientation are:

- loose coupling,
- service contracts,
- autonomy,
- abstraction,
- re-usability,
- composability,
- statelessness and
- discoverability.

These principles promise to increase agility by breaking up inflexible IT infrastructures, which are usually characterized by monolithic applications. The flexibility of such a SOA paradime, is laying its modular and decoupled development process, and in particular it's potential for application reuse enable enterprises to reduce their project risks and achieve a faster time-to-market [Bac00]. If applied correctly, the SOA principles will help to minimize the risks of enterprise IT by providing a sound architectural basis. Introducing such an architectural style will in general be a long-lasting process, and its beneficial effects will become apparent not all at once but steadily during this process [KBS06]. The challenge facing most companies is not whether to adopt SOA, but when and how to do so [WM06]. Following to market research institutes [ResSOA09], the SOA market is estimated at $3.3 billion in 2008 and is anticipated to grow at an average rate of 17.1% per year to $10.3 billion by 2015. The latest software releases of major vendors such as SAP, IBM, ORACLE (SUN, BEA), Microsoft etc. are mostly integrating SOA key principles or are at least web-service enabled (SAP with Netweaver, IBM with WebSphere, ORACLE with SOA Suite having integrated SUN and BEA AquaLogic, Microsoft with SOA BizTalk etc.).

The main objective of a SOA is consequently to provide mechanisms for allowing old and new technologies to be integrated and implemented dynamically by focusing into "business

services" rather than applications. Presentation, business logic and data are all on separate tiers and are loosely coupled, allowing the quick change of business processes. An organization can get new best-of-breed applications and integrate them easily with existing systems. A SOA is promoting reuse, so the time it takes to deliver new business functionality can be reduced [ATHEN03].

### 1.1.2. Why SOA

As described earlier, SOA is an architectural style using web services. This thesis will not focus in detail on technical specificities but more on the engineering method to implement such a SOA.

According to Blinco et al [BGLOS+09], it is difficult to define SOA as there are different perspectives on the SOA topic depending of the viewpoint taken. A business analyst will look at it as a set of services that the business wants to expose to their customers, partners, suppliers or internal process customers within the organization. A system architect looks more at the architectural style of service provider, service consumer, web-service descriptions and architectural principles addressing loose coupling, service contracts, autonomy, abstraction, re-usability, composability, statelessness and discoverability.

As the name tells, services play the major role. Actors can take the three different roles:

1. Service Registry
2. Service Provider
3. Service Consumer

Before understanding what happens between these three different roles, it is important to know what we understand as a service, IT service and finally also web-service.

The term "service" has strongly evolved from the early marketing-centric definitions in the 60ties to a more general understanding of services in the 80ties.The definitions of services vary on the different levels or topics they are related to.

**Definition 2: Service [QBP87]**

> *"Services include all economic activities whose output is not a physical product or construction, is generally consumed at the time it is produced, and provides added value in forms (such as convenience, amusement, timeliness, comfort or health) that are essentially intangible concerns of its first purchaser[QBP87]."*

New perspectives of services such as "systems design and operation" were defined by researches and are shifting over time.

**Definition 3: Service Systems [SMBG07]**

> *"a configuration of people, processes, technology and shared information connected through a value proposition with the aim of a dynamic co-creation of value through the participation in the exchanges with customers and external/internal service systems [SMBG07]."*

In the context of SOA, the definition of "web-service" is particularly important:

**Definition 4: Web-Service [KBS06]**

> *"A web-service is a software component of distinctive functional meaning that typically encapsulate a (high-level) business concept. It consists of a service contract, business logic, data and interfaces [KBS06]."*

A service encapsulates business logic within a distinct context. The context can be specific to a process or a business activity. Next, the business logic can include the business logic provided by other services, which is also called composition [Bri06].

Within a SOA system, services can be used by other services or other programs. In order to interact properly in-between them, these web-services must be aware of other services. Web-service descriptions (such as name, location and data exchange requirements of the service) explain exactly what the service is about to do. The manner in which services use service descriptions results in a relationship classified as loosely coupled. As we want services to interact meaningfully, they must exchange information. Therefore a framework called "messaging" can preserve their loosely coupled relationship [Erl05]. The description of a service is done with languages that are classified as platform specific (PSM) e.g. IDL (Interface Description Language) or WSDL (Web Service Description Language). A complete, independent and updated list of web-service standards can be found under [WIKI10b]. The following figure below is explaining what we will later define as SOA Heartbeat adapted from [Dos05]:

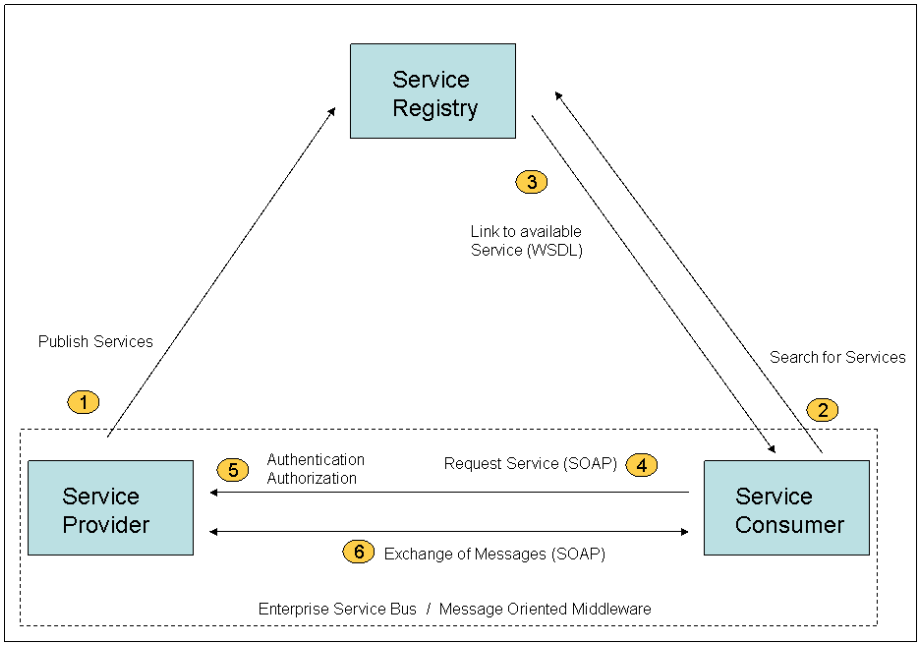



**Figure 1: Service Provider, Service Registry & Service Consumer**

The service provider creates a service. This is also known as "service deployment". To allow service consumers to use this service, it is necessary to publish it in a service registry (1). The service is stored in the registry and can be found through a search (2) from a service consumer, who wants to use this service. The registry tells the service consumer through the

service description that the service is available and indicates the related service provider, physical location, contact person, usage fees, technical constraints, security advice and available service levels (3). It is important to note two different types of registries: registries just for internal enterprise usage and registries for cross enterprise service integration, which have different requirements. The service consumer will then request the service from the service provider (4). The exchange of messages between service provider and service consumer is done with SOAP. Once, the service consumer has done the authentication and authorization, the service can be used by the service consumer (5).

The enterprise service bus connects the participants of a SOA – services and application front-ends – with each other. If two participants need to communicate, e.g. if an application front-end needs to invoke some functionality of a basic service, the service bus makes it happen. The service bus is not necessarily composed of a single technology. The main characteristics of a service bus are the following [KPS06]:

**Connectivity:** The primary purpose of the service bus is to interconnect the participants of a SOA. It provides facilities enabling the participants' application front-end and services to invoke the functionality of services.

**Heterogeneity of technology:** the service bus must embrace a variety of different technologies. The reality of enterprises is characterized by heterogeneous technologies. Consequently, the service bus must be able to connect participants that are based on different programming languages, operating systems, or runtime environments. The service bus must be able to support a multitude of middleware products and communication protocols.

**Heterogeneity of communication concept:** similar to the heterogeneity of technologies, the service bus must also embrace a variety of different communication concepts. Due to the divergent requirements of different applications, the service bus must enable different communication modes.

**Technical functionality**: the main purpose is primarily communication, but it must also provide technical services such as logging, auditing, security, message transformation, or transactions.

We finally conclude with the definition for SOA heartbeat as:

**Definition 5: SOA Heartbeat**

The "SOA Heartbeat" is defined as a number of processes and interactions taking place between service registry, service provider and service consumer with the objective to execute a web-service to fulfil a business requirement.

## 1.1.3. Problem Statement

### 1.1.3.1 SOA Challenges & Issues

A lot of organizations are facing the practical problem that they want to build a flexible, modular and easy to adapt IT environment to cope with business requirements. The SOA architectural style is promising advantages in comparison to traditional architectures. Grand challenges of the technical SOA and web-service aspects are resolved. But when is SOA

successful? Which promises have to be achieved? How can we measure SOA success against traditional architectures?

Early 2007, independent worldwide studies conducted by GARTNER [Pet07] with more than 1.400 Chief Information Officers (CIO), came to the result of decreasing importance of SOA for CIOs, while the level of SOA readiness and implementation did not progressed substantially SOA ranked only number 7 of Top 10 CIO priorities. The reason for this shift of priorities is linked mainly to two issues: first, the approach on implementation method and what abstraction models to use is very complex and today unclear. Second, the ROI of SOA can hardly be calculated. Recently, some research initiatives such as Mueller et al. [MVLR10] are on the way to explore the different categories of benefits (Organizational, Strategic, IT infrastructure, Operational and Managerial) with an economic potential model (SOA-EPM). Obviously, there is a complexity in this architectural style on how to achieve the promised benefits. This includes also the way to achieve a SOA implementation in an efficient and effective way. Which process is succesfull to implement such a SOA in an efficient and effective manner? This thesis will address with the SOA Method Engineering Framework the problem of complexity and also which process might be efficient and effective for an implementation of SOA.

The challenges materializing from the GARTNER study and other observation is also formulated within a service oriented computing research roadmap by Papazoglu, Traverso, Dustdar and Leymann [PTDL06]. They state: “What is required is a service-engineering method that allows enterprises to efficiently design and deploy services and which can easily embed changes into service-based applications at the rate and pace of change in the business design. It is from the correspondence that SOA deliver on the promise of more flexible business through a more flexible IT environment. This correspondence is represented as the service-oriented engineering methodology, in which business processes are modelled, analysed, assembled (possibly out of pre-existing components), deployed and monitored in a continuous and iterative manner.” Papazoglu and Dustdar are also doing research in this area, but they are not comparing different available SOA methods indicating strengths and weaknesses. Furthermore, concrete advice how to link strategy to processes and finally to the IT layer is missing. We will address this academic problem with the proposed SOA Method Engineering Framework.

This concrete need has been confirmed by a recently published work of identifying and classifying on-going SOA research. Viering et al. [VLA09] therefore name one of 4 SOA research types “How to design, implement, and manage SOAs?” *and* “specifying how organizations should apply the SOA concept, and might be most valuable from the practitioner’s point of view. It is associated with a constructivist type of research or design science, resulting in frameworks, reference models, methods, and management practices.” Consequently, the above mentioned challenges related to SOA implementation are still underserved and are still an original domain of SOA research. Consequently, we will ask ouerselves if and in what conditions the proposed SOA Method Engineering Framework will be successful or not.

Related research areas to the SOA Engineering Method are the following:

*Process-orientation:*
The “process-driven SOA” might be a possible solution for the open issue to resolve. Zdun and Dustdar [ZD06] describe the central challenge for the modelling of process-driven SOAs. Key issue is the integration of the different kinds of models and abstractions. This problem is

challenging, because so far there is no formal and precise modelling approach for integrating all kinds of models. The missing integration of process-driven SOA models for different levels of abstractions needs to be further analysed.

*Model-driven concept:*
In BPM as management discipline, many different languages and tools exist. The functionalities and characteristics are different and can lead to misunderstanding and failure. Furthermore, executable languages used to implement the models (e.g. process execution languages like BPEL [OASIS07], [IBMSS03] or programming languages) are also diverse. These identified issues are based on Model-Driven Software Development (MDSD) concept [SV05], which is a specialization of Model Driven Architecture (MDA) [Fra03] [OMG03]. The MDA defines an approach whereby new principles are promoted to separate the system functionality specification from its implementation on any specific technology platform. MDA is trying to achieve an architecture that will be language, vendor and middleware neutral [Fra03].

This approach is a contemporary approach to managing technology-independent service specifications, and implementing and managing SOA meta bus architectures by the Object Management Group (OMG) [KBS06]. The research issues in this context, based on Service – Oriented Modeling and Architecture [Sin07], are focussed on resolving mainly technical questions regarding service identification, service specification and service realisation.

*SOA Implementation Strategy*
The SOA implementation strategy can vary between different options such as for example "top-down", "meet-in-the-middle" and "bottom-up". Terlouw et al [TTJ09] have defined a so called "Delivery Strategy Assessment Method" (DSAM) determining the most appropriate SOA delivery strategy for an organization. In general we define Modeling Strategy for SOA delivery as follows:

**Definition 6: Modeling Strategy for SOA Delivery [PvdH06]**

> *"defines the approach that considers diverse business process realization scenarios evaluated in terms of costs, risks, benefits and ROI in accordance with business requirements and priorities [PvdH06]".*

Deliverables based on literature review and practical cases described in the INTEROP [BGBDK+05] and ATHENA [ATHEN05] – projects come to the result that a top-down method has more strengths than weaknesses. In our research, though not excluding other perspectives, we favour a top-down implementation strategy as changing to SOA must be motivated and supported from the IT strategy. The bottom-up strategy is coming from the web-service inventory and neglecting the business motivation for SOA. The first assessed SOA implementations in case studies [TTJ09] showed a clear tendency towards successful implementation if top-down strategy was selected.

These areas or principles are somehow related to the SOA Engineering Method question. SOA requires development discipline and methods that must be defined and enforced [Bar05]. Therefore, we will consider in the present work these principles in the context of the method definition with the objective to find out what role they play in practice and what decisions are linked to these principles.

### 1.1.3.2. Problem Statement

The previous section showed that there is a need for research in SOA method to develop a service engineering method as one of the identified grand challenges in service design and development. Since research has insofar been ineffective to cover this need, this still represents an original research domain. The academic proposals just include small parts of the target objective or miss important parts or use other principles as described in [Zim09] or [Tran09]. There are no proposals using an engineering approach as described and required in the SOA research roadmap [PTDL06]. Following to this, figure 2 is explaining the practical problem:

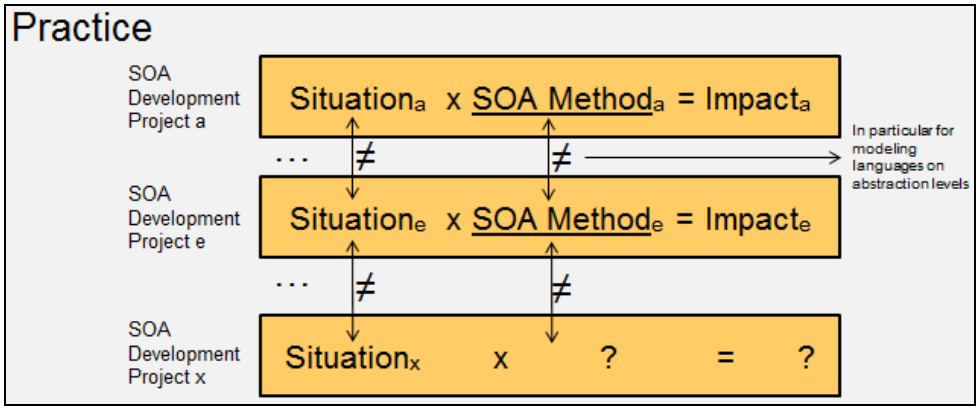



**Figure 2: Problem Statement in Practice**

Different SOA development projects exist in many organizations. These organizations are all in different situations meaning available systems, modelling tools, IT knowledge, scope, budgets etc. There exist many different available SOA Methods, which are proposed by various industrial service providers or academic researchers. These SOA Methods (list in chapter 2.5.) are defined following a specific viewpoint and specific concerns (details in chapter 2.1.). It is possible that for a given SOA development project and a specific situation the SOA method fits perfectly and delivers the expected result. Obviously the situation will differ in different SOA development projects as well as the used SOA Method e.g. the IBM SOA Method will be different from the ORACLE SOA Method, which will be different from the SAP SOA Method. One of the practical problems is that these SOA Methods do not necessarily fit to various situations (x). Also, modelling languages used in the SOA Methods are quite different on various levels. Furthermore, the integration between these modelling languages seems not to be trivial and also an original domain for research [ZD06].

Particular important is the focus on the modeling domain, which means a focus on modeling languages and also on what abstraction level to model and which modeling languages might be good candidates to be used in a SOA context. The present work will focus on this particular domain of modeling but in the context of a SOA Method application. There are certainly other relevant domains related to SOA concerning for instance web-service details. We use the terminology domain, a model, modeling as defined by Lankhorst [Lan05] explained in details through definitions 18,19 and 20 in the state-of-the-art (chapter 2.1.).

Next, the expected impact for the SOA development project of using a SOA Method in a particular situation might be positive or negative. This measurement is in practice very difficult, as project managers or CIOs cannot redo a project applying another SOA Method to compare. This behaviour would not be efficient in practice; therefore just lessons learned are recorded at the end of these SOA development projects.

## *1.2. Proposal & Research Questions*

### 1.2.1. Research Proposal

In order to resolve the identified challenges and problems, the following proposal is made:

First, the state-of-the art review should identify available SOA methods. It is important to understand the different viewpoints of these SOA methods and their degree of SOA topics coverage. The literature review is leading to the definition of a conceptual model (SOA Domain Model) taking different viewpoints and related SOA sub-domains. Based on this broad conceptual model the comparison of existing SOA method proposals is done. Furthermore, a feedback will be asked from practitioners' on available SOA methods, suited candidate modeling languages and the proposed SOA Domain Model. Finally, the need for resolving the academic and practical problem should be confirmed.

In order to find out, if a SOA Engineering Method Framework as claimed by academia can efficiently solve also the practical problem in organizations, method engineering principles should be used and linked to the SOA Domain Model. Therefore, method fragments should be created and formalized following method engineering principles. These fragments are compiled in a configuration process for SOA situational method using a tool to prototype the approach and allow efficient re-use in field trials for real-life application and will allow investigation whether practical efficiency problems in method application could be removed.

Additionally, the principles of model-driven architecture and process-orientation will be investigated if they are considered as important principles in this context. Related to these principles, we aim to provide potential candidate notation and an attempt of suitability related to SOA Method Engineering Framework.

Based on the problem statement the following SOA Method Engineering Framework (SOA-MEF) is proposed:

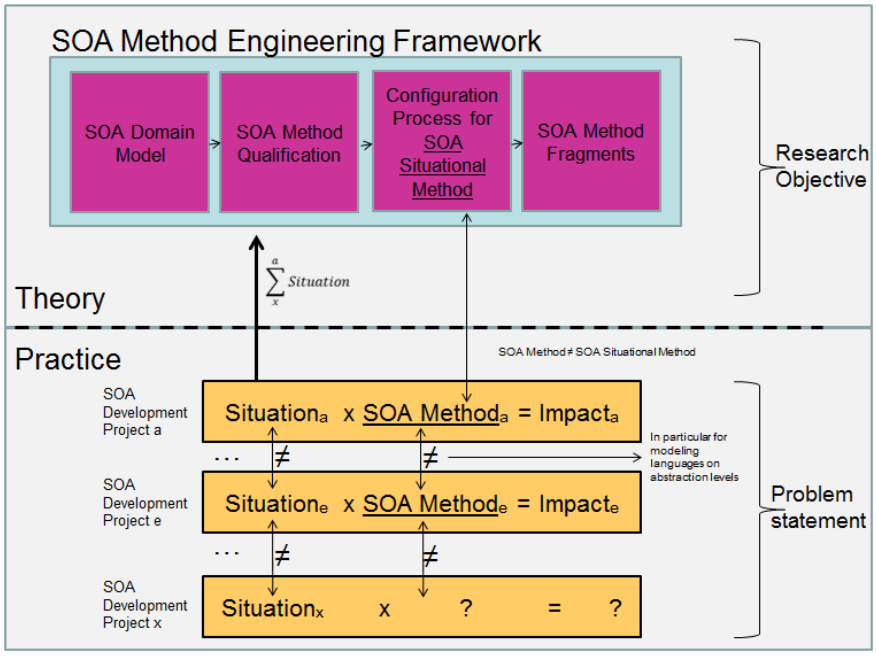



**Figure 3: Research Objectives coping to Problem Statement**

The SOA-MEF is decomposed into 4 main artifacts:

Firstly, a model (here: SOA Domain Model) is required to summarize and generalize various aspects (here: SOA sub-domains) that must be covered in a model-driven and process-oriented SOA development project. The SOA Domain Model should summarize contents of available SOA Methods. This model may indeed narrow the communication gap that often exists between managers (business) and developers (IT) by providing them a common reference framework, concepts and vocabulary. This is particularly done by describing SOA Domains which include sub-domains and activities. Domains can be considered as clusters (e.g. SOA modelling) with sub-domains (e.g. SOA modelling notation). Each sub-domain includes a series of activities (e.g. model business requirements with BPMN [OMG09]).

Secondly, it is required to think about a possibility to get an idea what range and how deep available SOA Methods are covering SOA domains. SOA Methods are very broad and the before identified domains and related SOA sub-domains could be a way to compare in a structured way SOA Methods and identify areas of strengths and weaknesses of these available SOA Methods.

Third, with the help of Method Engineering (details in chapter 1.2.1.1.) the necessary academic deepness is achieved by proposing a *situational* SOA Method. Other than the SOA Methods, the situational SOA Method is able to adapt to particular situation. A **C**onfiguration **P**rocess for SOA **S**ituational **M**ethod (CP-SOA-SM) should in general be able to describe the process of creating fragments and applying them in a specific situation. The problem of non-fitting SOA Methods to a specific situation can be avoided.

To achieve this, SOA method fragments should be created and formalized from available SOA Methods. With these SOA Method Fragments, a perfectly fitting SOA Method could be configured as only the relevant fragments are chosen that cope with situation for concrete SOA definition project.

Due to space restrictions and also the range of complexity in other domains, it is important to emphasize that the focus of this thesis is lying on the details of the SOA modelling domain.

It is important to mention that “evaluation” in this thesis does not mean automatically coping to deep statistical evaluation requirements, but to consider the evaluation more on a qualitative level. Furthermore, the qualitative validation is only a first cycle of many iterations that could be done. As the proposed topic and the SOA Method Engineering Framework is so broad and complex, more iterations have to be done to achieve “evaluation” with results which can be generalized. What can be understood by “evaluation” of the different proposed artifacts is detailed in every relevant chapter.

### 1.2.1.1. Method Engineering (ME)

In order to cope with the “Engineering” dimension for methods, the thesis will apply Method Engineering (ME) principles. In order to engineer a SOA implementation method, the definition of ME concept developed by Harmsen and Saeki [HS96] as well as Brinkkemper, Lyytinen and Welke [BLW96] is used: “Method engineering in the field of information

systems is the discipline to construct new methods from existing methods. It focuses on the design, construction and evaluation of methods, techniques and support tools for information systems development." Furthermore, according to Roland [Rol08] "method engineering wants to improve the usefulness of systems development methods by creating an adaptation framework whereby methods are created to match specific organizational situations." Here, the objective lies in the demonstration that SOA method fragments can be formalized.

#### 1.2.1.2. Tooling

In order to demonstrate that the generated artifacts can be used and applied in case studies, it is required to think about a tool, which can be used to ensure effective and efficient structure and application of CP-SOA-SM. The tool used can also be summarized as a prototype with the objective to demonstrate the applicability of artifacts in real-life cases.

Following to Satzinger, Jackson and Burd, [SJB09] "prototyping is the process of building a model of a system. In terms of an information system, prototypes are employed to help system designers build an information system that is intuitive and easy to manipulate for end users." and "The prototypes are not built for full functionality but are built to see if the prototypes are feasible for what goals the business is trying to achieve." As a prototype tool, it seems that extensible process engineering tools can support the described objective in that context. For example, such a tool should provide method authoring, process authoring, library management and configuration functionalities [Eclipse09].

With method authoring we understand the possibility to capture re-usable method building blocks with an underlying meta model. The method blocks should be re-used through inheritance-type relationships.

Process authoring ability is about construction of reusable method fragments in processes may for example define how to create BPMN model. This method fragment can now be reused in a variety of processes and delivery processes.

The tool should provide a database allowing flexible configuration of method fragments, XMI-based exchange format, packaging of method fragments and processes into plug-ins for exchange to other databases. To summarize this, the tool should integrate ME principles, freely available, use open standards and extendable to other process engineering tools.

### 1.2.3. What the PhD is not…

After defining the contribution, it is important to mention what is out of scope for this thesis:

- Proposing a modeling language or method for SOA or to improve modeling languages (this is addressed in the PhD of Stein [Stei09].)

- Investigating into technical details of SOA or technical issues of SOA related to the implementation method [Gün05] or web-service problems related to semantics, ontologies etc.

- The creation of a SOA meta-model in the domain of technical implementation. SOA meta models already exist by [MacK06] [But09] or the OMG SoaMl [OMG09b].

- Achieve full statistical validation of results. We are proposing a first cycle with more iterations to be done in future work.

### 1.2.2. Research Questions

Our research contribution aims at proposing a SOA Method Engineering Framework to respond to the research need formulated in the SOA Roadmap by Papazoglu, Traverso, Dustdar and Leymann [PTDL06]. Second, the research need is also responding to a practical problem of using an efficient method which can be adapted to the situation and the context of organizations.

As explained in the problem statement and addressed in the research proposal, the following questions are posed:

***Q1.: How can differences of available SOA Methods be identified and characterized?***

***Q2.: What is required for a method which is situated?***
***Q2.1.: What is required to decompose a SOA method?***
***Q2.2.: What is required for recomposing a SOA method***

***Q3.: How can the configuration process for SOA situational Method support the decisions taken in practice by organizations?***

***Q4.: Which candidate modelling languages are suited to serve in the SOA implementation context and on what level of abstraction?***

***Q5.: How to integrate the different kinds of models in a single method?***

***Q6.: What about the quality of generated SOA Method and the achieved results out of SOA Method?***
***Q6.1.: Is the quality of generated SOA Engineering Method satisfactorily?***
***Q6.2.: Is the achieved result from SOA Engineering Method satisfactorily?***

## *1.3. Scope of Work*

### 1.3.1. Research Design

After having defined the different types of problems to address, we need to propose a research method.

Hence, as Jick [Jic79], Mingers and Broclesby [MB97] and Blaikie [Blai05] argue, research methods should be combined meaning to gather quantitative and qualitative data. In the specific context of the present research, this might provide a research design to allow a more holistic study and validation of the research questions. Furthermore, they argue that experiments may not fit within the proposed research design as experiments need to take

place in a controlled environment. In order to enrich the proposed research method, we will use exploratory research style elements (without going too far as this would be a thesis for its own) with the objective to collect information for a better understanding of the problem of SOA complexity and the motivation by practitioners to implement SOA. The perceived problems both from academia and also from practicioners should be taken as an input helping to better and more precicely formalize the research design mix in table 1. It is not the objective to prepose a complete exploratory research cycle before the design science method mix, but to embed literature review and also a practicioners survey into the method mix.

If we follow the research framework outlined by March and Smith [MS95] combining research activities and research outputs, we can show the scope and method mix of the thesis:

**Table 1: Scope and method mix for thesis**

<table>
<tr><td>Research Output (Artefacts)</td><td colspan="3">Research Activities (Theorize and Justify are out of scope)</td></tr>
<tr><td></td><td></td><td>Build</td><td>Evaluate</td><td>Research-Questions</td></tr>
<tr><td rowspan="4">SOA sub-domains<br><br>SOA Domain Model<br><br>SOA Fragments & Process<br><br>Result from Instantiation</td><td>Con-structs</td><td>Literature Review Artifact:<br>“SOA sub-domains for SOA Method Implementation”</td><td>Survey Artifact:<br>“Feedback on SOA Complexity by Exploratory Survey”<br><br>“Testing SOA domains and sub-domains”</td><td>Q1 and Q4<br><br>Q1 and Q4</td></tr>
<tr><td>Models</td><td>Literature Review Artifact:<br>“SOA Domain Model”</td><td>Survey Artifact:<br>“Testing of SOA Domain Model”</td><td>Q2 and Q5</td></tr>
<tr><td>Methods</td><td>Method Engineering Artifact:<br>“Configuration Process for SOA Situational Method” and „Method Fragments“</td><td>Not possible</td><td>Q3 and Q6</td></tr>
<tr><td>Instanti-ation</td><td>Method Engineering Artifact:<br>“Application of SOA Method Engineering Framework on two application cases”<br><br>“Prototyping of a Tooling Support” for SOA Method Engineering Framework”</td><td>Application case Artifact:<br>“Evaluation of SOA Method Engineering Framework on two application cases”</td><td></td></tr>
</table>

March and Smith [MS95] state natural science tries to understand reality, whereas design science attempts to create things that serve human purposes. Rather than producing general theoretical knowledge, design research produces and applies knowledge of tasks or situations

in order to create effective artefacts. Its products are of four types, i.e. constructs, models, methods, and instantiation. In our research, the artefact is a framework including a SOA Domain Model, SOA Methods Evaluation, a Configuration Process for SOA Situational Method and SOA Method Fragments. The instantiation of this artefact will be built through the model and applied in two real-life case studies. This evaluation of the proposed artefacts will allow refinement and practical inside how to apply the artefacts.

Research outputs or artifacts cover constructs, models, methods and instantiations [MS95]:

"Constructs are the concepts, vocabulary and conceptualizations that are used to describe, think about and solve problems and tasks within a given domain.

Models are a set of propositions or statements that integrate a number of construct by expressing the relationships that exist among them.

Methods are a set of steps, algorithms or guidelines used to perform a task. It is based on a set of underlying constructs and a representation (model) of the solution space.

Instantiations are realizations of artifacts in its environment. They operationalize constructs, models and methods and demonstrate their feasibility and utility.

Research activities comprise building, evaluating, theorizing on and justifying artifacts. The two former activities are the domain of design science, whereas the two latter are the domain of natural and social sciences.

Building refers to the conception and construction of viable and purposeful artifacts – in the form of constructs, models, methods and instantiations - aiming at resolving a problem. Their successful development demonstrates their feasibility.

Evaluating refers to the assessment of the proposed artifacts according to suitable metrics. The relevance and contribution of a specific design science research artifact is generally judged on the basis of its value or utility to a community of users, be it for its novelty, increased efficiency or effectiveness compared to existing artifacts.

Theorizing refers to the construction of theories that try to explain how or why some phenomena of interest happen.

Justifying finally refers to proving that theories are truthful through the gathering of empirical scientific evidence that supports or refutes them."

Note that a more detailed discussion of the applied research methods proposed at each stage is provided at the beginning of each corresponding chapter.

## 1.3.2. Research Structure

The research is decomposed into the following chapters:

**Table 2: Decomposition of Research into Chapters**

<table>
<tr><td rowspan="5">Chapter 1<br>Introduction</td><td></td><td>Build</td><td>Evaluate</td><td rowspan="5">Chapter 7:<br>Conclusion</td></tr>
<tr><td>Construct</td><td rowspan="4">Chapter 2<br>Literature Review<br><br>Chapter 3:<br>Contribution:<br>1.) SOA Domain Model,<br>2.) SOA Method Qualification<br><br>Chapter 4:<br>3.) Configuration Process for SOA Situational Method<br>4.) SOA Method Fragments<br><br>Chapter 5:<br>SOA Tooling & Prototyping, Implementation</td><td rowspan="4">Chapter 3:<br>Qualitative survey to evaluate SOA Domain Model and SOA Methods<br><br>Chapter 6:<br>Multiple field trials to apply, evaluate and refine Configuration Process for SOA Situational Method</td></tr>
<tr><td>Models</td></tr>
<tr><td>Methods</td></tr>
<tr><td>Instantiation</td></tr>
</table>

In chapter two we review the state-of-the-art about SOA Methods but also to related topics like Enterprise Architecture (EA), modeling languages, interfaces and translation between models. EA has been chosen as starting point, because the „helicopter view“ on strategy, processes and IT is addressed by EA. It is therefore important to understand first what different components, views and relationships are important in the context of SOA Method. With the state-of-the-art analysis, available SOA Methods should be summarized and evaluated where different levels of abstractions and related modeling notations suited for SOA implementation are of particular interest. The identified issues on current SOA implementation methods will confirm the relevance of the research.

Chapter three is about the detailed construction of the SOA Domain Model which is a conceptual modeling exercise. Based on the state-of-the-art elements gathered, sub-domains are classified, structured, condensed and finally a SOA Domain Model is proposed. Under the chosen angle of ***T***op-***D***own, ***M***odel-***D***riven and ***P***rocess-***O***riented (TD-MD-PO) SOA implementation method, there will be a link to notations and process modeling languages, as this is the way to abstract from the complex reality a model allowing concentrating on details important for SOA Method. Therefore, a focus will be done on the domain “SOA Modeling”. Then, the available SOA Methods are evaluated with the SOA Domain Model.

Chapter three is also dedicated to the survey that has two objectives: First the survey should test the proposed SOA Domain Model by experienced practitioners and second gather knowledge on important questions e.g. used and successful modeling notations and the degree of satisfaction with available SOA Methods.

Chapter four is detailing the configuration process for SOA situational method. Next, SOA Method fragments are created from available SOA Methods. Two standard SOA Methods are formalized with ME into method fragments referring to the SOA Domain criteria's of modeling.

In chapter five, existing tools are used to create a prototype supporting the approach. This way, the conceptual work from chapter two and especially chapter three with the research contribution is implemented and principles such as ME are enforced. Method fragments are created and stored in a method engineering tool and are then ready for re-use in application cases chapter six.

Chapter six describes the application of the CP-SOA-SM to 2 field trials. The objective is to apply the SOA Engineering Framework with the main artifacts of SOA Domain Model, SOA Method Fragments and Configuration Process for SOA Situational Method. The field trials should demonstrate applicability of the assembled generic method and show in detail the design rationales or decisions that have been taken for the specific implementation examples. To mitigate the risk of generalizability [Ben87] which represents a weakness of case study method, both cases have followed the same process using the same guidance documentation.

Chapter seven concludes this thesis by making a summary of its various contributions and by proposing further interesting research directions.

# CHAPTER 2

# STATE-OF-THE-ART



Chapter two is focussing on literature review for the posed research questions in the introduction specifically related to a top-down, model-driven and process-oriented viewpoint (Section 2.1. to 2.4.) and a SOA Engineering Method (section 2.5 and 2.6.). Consequently, the chapter is starting with Enterprise Architecture (section 2.1.) because the conceptual

foundation and understanding for the SOA Domain Model is built. Modelling languages are an essential part of a model-driven and process-oriented SOA implementation and therefore identified on each level of abstraction (section 2.2.). Next, the interfaces of conceptual modelling (section 2.3.) between the abstraction layers and how these modelling notations can be transformed or interfaced (section 2.4.) is described. Available SOA Methods are introduced (section 2.5.) and first SOA sub-domains are identified. All available and current proposals are listed and a preliminary grouping of relevant SOA sub-domains covered in the methods is presented. Method Engineering is introduced (section 2.6.) as an engineering approach for methods, which needs to be applied in this present work to allow situational and context tailored method application. At the end of this chapter, a summary recaps the main conclusions (Section 2.7.) of literature review.

## *2.1. Why Enterprise Architecture is the starting point*

EA is about essential aspects of an organization that can be formalized and shown. These include but are not limited to [JV90] [VZ93]:

- Enterprise functionality and behaviour in terms of processes, activities, operations and triggering events;
- Decision-making processes, decision flows and decision centres';
- Physical components or resources, e.g. machines, tools, networks; applications;
- Business data and information with the related flows in the form of orders, documents, data items, data files, databases
- Enterprise knowledge and know-how, domain and industry specific knowledge, rules, management guidelines, policies and procedures
- The organisation with human individuals with divisions, decision levels, roles & responsibilities, knowledge and competencies.

EA provides a way to handle complexity of modern information-intensive companies or organizations. Therefore, IT divisions need ways to express architectures as clearly as possible to make sure a common understanding between IT and business can be realized. The link to the research question is about a framework (here SOA Method Engineering Framework) to apply including methods and models. These models are used on different levels of abstractions. As the SOA method qualification includes many aspects, it is necessary to start with EA concepts. EA is relevant for the research topic, as terminology such as "Concern", "Architecture Viewpoint", "Architecture View", "Architecture Model", Domain", "Modelling", "Model" etc. are key for the research contribution as introduced in the first chapter.

Actually, it often appears that in organizations no common understanding on terminology can be achieved and finally communication is rather hard [Lan05]. Also, many different definitions on EA can be found in different books. Architectural frameworks structure architectural description techniques by identifying and relating different architectural viewpoints and the modeling techniques associated with them. They typically define a number of conceptual domains or aspects to be described [Jon02]. For instance, an architecture principles catalogue [GP11] with 59 architecture principles is allowing organizations to structure the complexity and enhance the design of organizational set-up.

This thesis will not focus on comparing the different EA frameworks as it is quickly evolving and this type of analysis has already been done several times e.g. in EA books [Lan05], [LPWCS09],[GP11],[SS93],[RWR06],[Ver96], important EU-funded projects [ATHEN04], [BGB+05] or describing famous Architecture Frameworks in detail such as:

- 4+1 View Model of Architecture [Kru95]
- AKM technology and Knowledge Space method from Computas [ATHEN04]
- ArchiMate [Lan05]
- ARIS [Sch93]
- CEN ENV 40 003 [Cen90] [Cen95]
- DoDAF/C4ISR (The Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework) [C4IS97]
- GERAM Generic Enterprise Reference Architecture and Method [BN94]
- GRAAL framework (Guidelines Regarding Architecture Alignment) [EBFG+02], [WBFG03]
- GRAI/GIM [VCZD91].
- ISO/IEC/IEEE 42010 standard [ISO11].
- Model Driven Architecture (MDA) [Fra03], [OMG03]
- Nolan Norton Framework [Zee00].
- PERA [Wil92]
- RM-ODP (Reference Model for open distributed processing) [Iso00a].
- RUP: Rational Unified Process [Fra03] [WBFG03]
- SEAM: Systemic Enterprise Architecture Method [Weg03]
- TEAF Method from the US Dept. of Commerce [ATHEN04]
- The Four-Domain-Architecture [IG04].
- The Zachman Framework [Zach87], [SZ92]
- TOGAF [Ope02].
- TOVE [Fox92]

We will refer to the definition of IEEE [IEE00] which has been further developed together with ISO and IEC into the ISO/IEC/IEEE 42010 standard [ISO11]. For the definition of this standard, ISO has conducted a survey [ISO10] on 52 architecture frameworks. This list of architecture frameworks can be probably considered as one of the most exhaustive enumerations. Shortcomings on the initial IEEE definition [IEE00] identified by Lankes et al. [LMW05] on management views and software maps have been addressed in the latest reworked version of ISO [ISO11] as presented in the next section.

### 2.1.1. Conceptual Foundation for Architecture through the ISO/IEC/IEEE 42010 Standard

The standard ISO/IEC/IEEE 42010 [ISO11] specifies terminologies which is relevant for the research topic and defines architecture framework and requirements on architecture frameworks. This is also including architecture descriptions of systems, architecture description languages (ADL) and architecture viewpoints (AV). An Architecture Framework (AF) is defined as follows:

**Definition 7: Architecture Framework [ISO11]**

> *"Conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders." [ISO11]*

Figure 4 explains the context of conceptual architecture with an existing system, which is situated in its environment. Following to ISO [ISO11], that "Environment" could include other "Systems". "Stakeholders" have interests (here: "Concerns") in a "System". A "Purpose" (earlier version of the standard: mission) is one very common "Concern". "Systems" have "Architectures" and "Architecture Description" is used to express "Architectures".

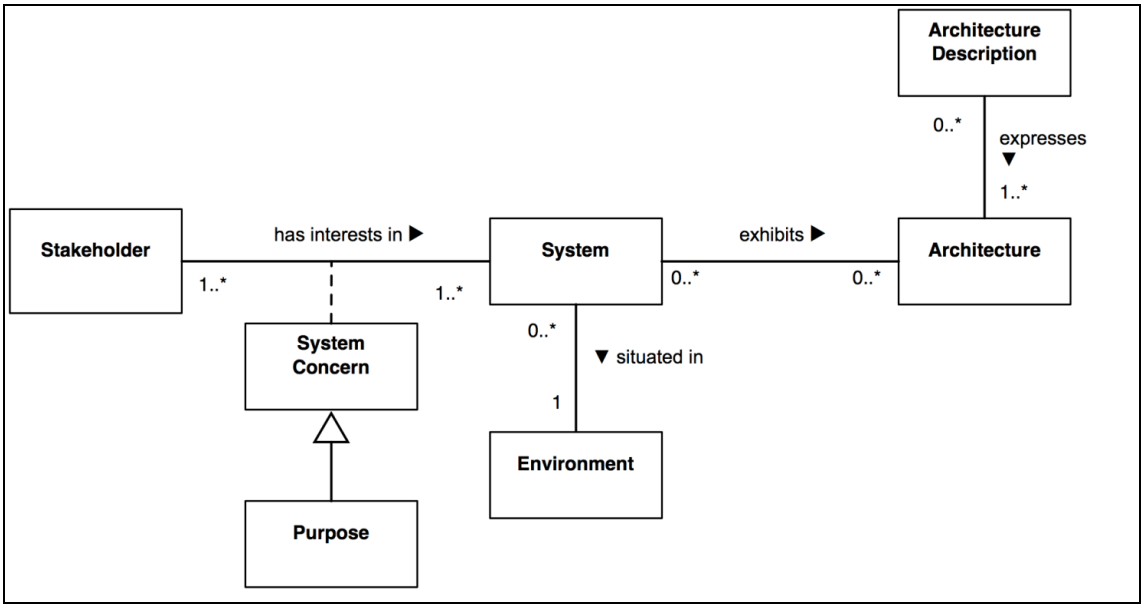



**Figure 4: Context of Architecture [ISO11]**

**Definition 8: System [ISO11]**

> *"system is used as a placeholder – e.g., it could refer to an enterprise, a system of systems, a product line, a service, a subsystem, or software. Systems can be man-made or natural. Nothing in the Standard depends upon a particular definition of system. Users of the Standard are free to employ whatever system theory they choose." [ISO11]*

**Definition 9: Environment [ISO11]**

> *"Every System exists in its Environment. A System acts upon that Environment and vice versa. A System's Environment determines the range of influences upon the system. In the Standard, Environment is intended in the widest possible sense to include operational, developmental, regulatory, and all other influences which can affect the architecture. These influences are captured as Concerns." [ISO11]*

Systems have Architectures. In the Standard, the architecture of a system is defined as:

**Definition 10: Architecture [ISO11]**

> *"fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution." [ISO11]*

Following to this, architecture description has to be defined:

**Definition 11: Architecture Description [ISO11]**

> *"An Architecture Description is an artifact that expresses an Architecture. Architects and other Stakeholders use Architecture Descriptions to understand, analyze and compare Architectures." [ISO11]*

The following diagram depicts the content of an architecture description and the relation between those content items when applying the standard to express the architecture for some system of interest [ISO11]:

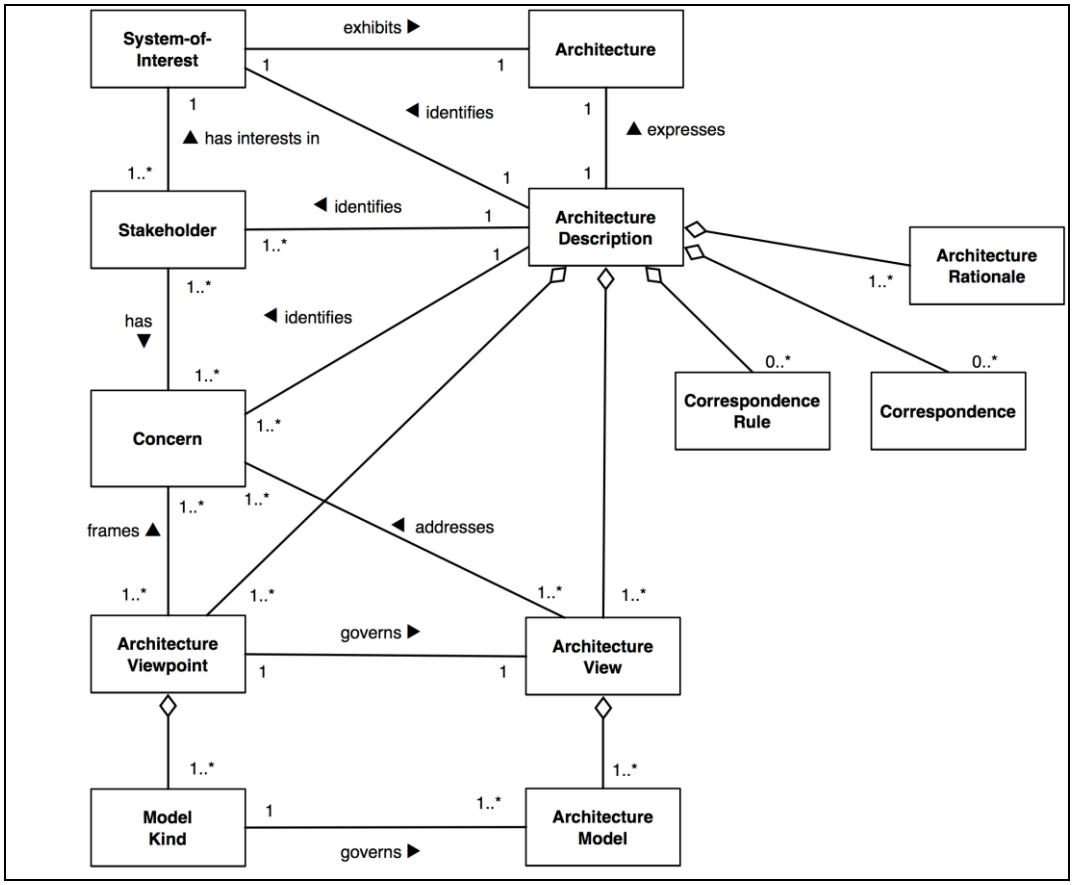



**Figure 5: The Core of Architecture Description [ISO11]**

**Definition 12: Stakeholder [ISO11]**

> *"Stakeholders are individuals, groups or organizations holding Concerns for the System of Interest. Examples of stakeholders: client, owner, user, consumer, designer, maintainer, auditor, certification authority, architect." [ISO11]*

**Definition 13: Concern [ISO11]**

> *"A Concern is any interest in the system. The term derives from the phrase "separation of concerns" as originally coined by Edsgar Dijkstra. Examples of concerns: (system) purpose, functionality, structure, behavior, cost, supportability, safety, interoperability." [ISO11]*

**Definition 14: Architecture Viewpoint [ISO11]**

> *"An Architecture Viewpoint is a set of conventions for constructing, interpreting, using and analyzing one type of Architecture View. A viewpoint includes Model Kinds, viewpoint languages and notations, modeling methods and analytic techniques to frame a specific set of Concerns. Examples of viewpoints: operational, systems, technical, logical, deployment, process, information." [ISO11]*

**Definition 15: Architecture View [ISO11]**

> *"An Architecture View in an AD expresses the Architecture of the System of Interest from the perspective of one or more Stakeholders to address specific Concerns, using the conventions established by its viewpoint. An Architecture View consists of one or more Architecture Models." [ISO11]*

**Definition 16: Architecture Model [ISO11]**

> *"A view is comprised of Architecture Models. Each model is constructed in accordance with the conventions established by its Model Kind, typically defined as part of its governing viewpoint. Models provide a means for sharing details between views and for the use of multiple notations within a view." [ISO11]*

**Definition 17: Model Kind [ISO11]**

> *"A Model Kind defines the conventions for a type of Architecture Model." [ISO11]*

The terms "Architecture Rationale", "Correspondence", "Correspondence Rule" are not further defined as these terms are not relevant for the further research.

### 2.1.2. Conceptual Foundation for SOA Domain Model

Following to Lanckhorst [Lan05], analysts and modellers may decide to zoom into a particular part of the universe they observe. Then, they will zoom into a particular part of their conception of the universe, here the enterprise. Related to the communication of actors, different terms need to be defined. Consequently, a domain needs to be defined as

**Definition 18: A Domain [Lan05]**

> *"A Domain is any subset of a conception (being a set of elements) of the universe that is conceived of as being some "part" or "aspect" of this universe."[Lan05]*

In this context, we need to clarify the definition of a model:

**Definition 19: A Model [Lan05]**

> *"A Model is a purposely abstracted and unambiguous conception of a domain." [Lan05]*

**Definition 20: Modelling [Lan05]**

> *"Modelling is the activity of purposely abstracting a model from a part from the universe." [Lan05]*

**Definition 21: A View [Lan05]**

> *"A View is a representation of a whole system from the perspective of a related set of concerns." [Lan05]*

**Definition 22: A Viewpoint [Lan05]**

> *"A Viewpoint is a specification of the conventions for constructing and using a view; a pattern or template from which to develop individual views by establishing the purposes and audiences for a view and the techniques for its creation and analysis." [Lan05]*

We will base terminology for the artefact of conceptual "SOA Domain Model" on the terminology defined by Lankhorst.

### 2.1.3. OMG Model-Driven-Architecture

A well-recognized approach to classify different types of models is the MDA developed by the Object Management Group [OMG03]. The objective is to provide an open, vendor-neutral approach of interoperability. It builds upon the Object Management Group's modeling standards: the Unified Modeling Language (UML) initially developed by [JBR99], the Meta Object Facility (MOF), and the Common Warehouse Meta-Model (CWM). Platform-independent application descriptions built with these standards can be realized using different open or proprietary platforms, such as CORBA, Java, .NET, XMI/XML and Web Services. Currently, the MDA paradigm could fundamentally change the way in which software is developed. MDA aims at raising the level of abstraction at which software solutions are specified by defining a framework supported by a collection of standards that sets a standard for generating code from models and vice versa. Kent [Ken02] is summarizing MDA as "guidelines which is focusing on architecture, on artifacts, on models. It aims to exploit the usefulness of models as tools for abstraction, for summarizing and for providing alternative perspectives. […] A clear goal is that transformations between models should at least be partially automated, thereby reducing the burden of keeping models benefits in balance with the cost of their maintenance". Next, we introduce shortly the different MDA abstraction models:

The Computation Independent Model (CIM) represents requirements for the systems by describing the situation in which the system will be used. Such a model is sometimes called a domain model or a business model and hides information about the use of automated data processing systems [Lan05].

The Platform-Independent Model (PIM) describes the operation of a system while hiding the details necessary for a particular platform. The model focuses on specifications that are not changing from one platform to another.

A Platform-Specific Model (PSM) combines the specifications in the PIM with the details that specify how these systems are using a specific type of platform [Lan05] e.g. CORBA.
Figure 6 is illustrating the matching between the MDA method abstraction levels and the higher grained abstraction levels of Strategy, Processes and IT.

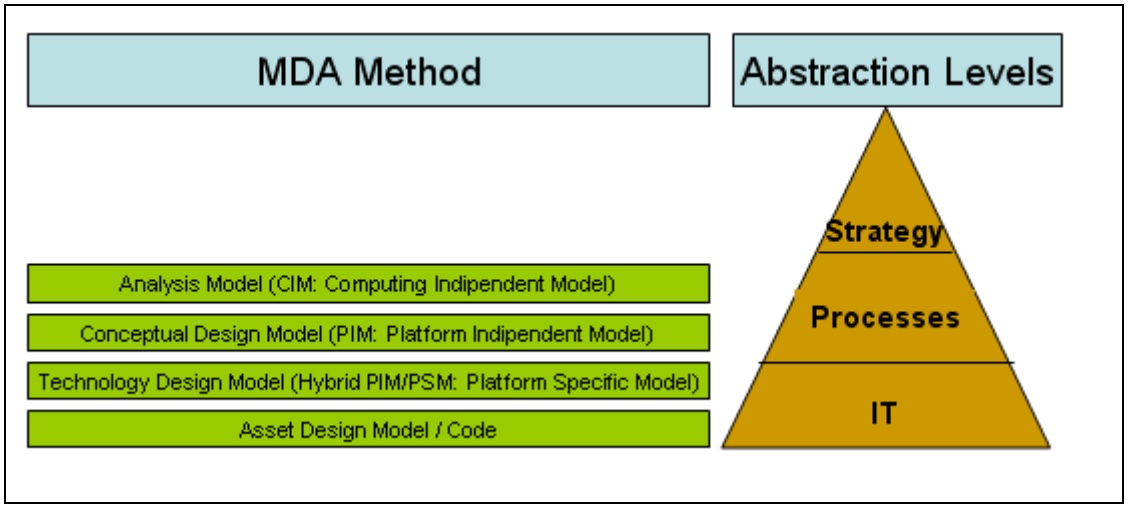



**Figure 6: Matching of MDA Models vs. Abstraction Levels**

UML is considered as the "de facto" modeling language for both PIMSs and PSMs. The first reason is the fact that it is the modeling language developed by the OMG [OMG01]. Second, UML is considered as a "semantically rich" language [Fra03]. This means that based on a meta-model, the objects used are semantically defined and allow a translation from the PIM view to the PSM view [PM06]. In the same context, Kleppe et al. [KWB03] distinguish between well-defined languages and not well defined languages. Following to [KWB03], UML is "a well-defined language because of form (syntax), and meaning (semantics), which is suitable for automated interpretation by a computer".

At the CIM level, it is more complicated as we have the notion of different views. This issue is explained in the "4+1" views on architecture design defined in the Rational Unified Process (RUP). Following to Kruchten [Kru95], the "4 + 1 View Model" describes "software architecture using five concurrent views, each of which addresses a specific set of concerns: The logical view describes the design's object model, the process view describes the design's concurrency and synchronization aspects; the physical view describes the mapping of the software on hardware and shows the system's distributed aspects, and the development view describes the software's static organization in the development environment. Software designers can organize the description of their architectural decisions around these four views and then illustrate them with a few selected use cases, or scenarios, which constitute a fifth view." The architecture is partially evolved from these scenarios.

MDA has been developed as a new philosophy for the software development (based on object oriented design). Nevertheless the principles could also be used for a SOA implementation if a semi-automatic and automatic transformation from one level (CIM, PIM, PSM) to another should be realized. The separation of modelling languages corresponds to the separation of concerns at the architectural level. Therefore languages are needed on the different levels of abstraction to describe functional (business requirements) and non-functional characteristics (transactional behaviour, security and persistence).

### 2.1.4. Summary on Enterprise Architecture context

The Enterprise Architecture Framework is the starting point for the analysis of available methods for SOA implementation and the underlying modelling techniques. As described, many different frameworks exist, whereas in all frameworks modelling related to a conceptual level plays a key role.

## *2.2. Modelling Languages in the context of SOA developments*

### 2.2.1. Enterprise Modelling

The term "Architecture Model" has already been defined in the context of the latest Architecture Framework [ISO11], but Enterprise Modelling (EM) has different roots and is rapidly introduced.

Historically, this journey started in the 70ties with research on database design with models like the "Entity Relationship Model" [Che76]. During 80ties, large Computer Integrated Manufacturing (CIM) [Sch83] projects started e.g. ICAM (Integrated Computer Aided Manufacturing) led by the US Air Force or CAM-I (Computer Aided Manufacturing – International). Early 90ties, first software appeared offering toolboxes for process modelling. These tools became more and more mature and offered new modules or completely new software to manage "workflow", meaning the automated execution of processes. Most of these process modelling tools disappeared again, and new, based on new customer requirements, were created and marketed. Beginning of the 20$^{th}$ century, the big back-end software companies well-known for their ERPs (SAP, ORACLE) or platforms (IBM, Microsoft, SUN) started to integrate processes into their software.

Following to Vernadat [Ver96], modeling is looking at the what, how, when and who aspects of an enterprise.

**Definition 23: Enterprise Modeling [Ver96]**

> *"EM is the set of activities or processes used to develop the various parts of an enterprise model to address some desired modelling objectives." [Ver96]*

and

*"EM can also be defined as the art of "externalising" enterprise knowledge, i.e. representing the enterprise in terms of its organisation and operations (e.g. processes, behaviour, activities, information, object and material flows, resources and organisation units, and system infrastructure and architectures)." [Ver96]*

### 2.2.2. Basic Modeling Principles

Ross and Schonman [RS97] identified 15 main principles for modelling techniques:

- the definition of the purpose of the model
- the range of the model, i.e. the scope or domain covered by the model (also called the universe)
- The viewpoint on the model, i.e. which aspects are covered and which are left out by the model, and
- The detailing level of the model, i.e. the level of precision or granularity of the model regarding the reality modelled. Obviously, the degree of model details depends on the way the observer understands the reality.
- Separation of concern; i.e. due to the huge complexity of a company, it would not be realistic to analyse it as a whole, but to decompose in pieces corresponding to a functional area or domain.
- Functional decomposition, i.e. decomposition from high level down into details. Ross calls it stepwise-refinement. Meanwhile it is called "drill-down" [Sche93], "decomposition" IDEF3 [MCFKP+95] or "micro-macroflow". [ZD06]
- Modularity should allow to plug&play with the different building blocks used in a model.
- Model genericity means the definition of building blocks into classes with individual attributes and metadata.
- Reusability of building blocks, i.e. organizational units defined in the Organizational Chart is reused in flow models to show who is performing the activity.
- Conformity needs to be addressed to make sure the syntax and semantics or also called modelling conventions are met.
- Visualization of models help to communicate
- Simplicity versus adequacy means finding the right balance between the expressions needed to achieve to show the correct content and the richness of the language and the effort to learn and master it adequately.
- Principle of management and complexity is related to the ability of representation of systems with great complexity.
- Principle of rigor of representation means that the model must neither be ambiguous nor redundant nor serve as a basis for verifying properties, analysing behaviour or simulating the system modelled.
- Data and control flow need to be separated because control flow is triggered by events. However, data plays an important role and needs to be considered as input or output of activities.

Following to ISO 19439:2006 for Enterprise Modelling [ISO06b], the standard "serves as a common basis to identify and coordinate standards development for modelling of enterprises, emphasising, but not restricted to, computer integrated manufacturing. ISO 19439:2006 also serves as the basis for further standards for the development of models that will be computer-enactable and enable business process model-based decision support leading to model-based

operation, monitoring and control…" and "…four enterprise model views are defined which are: function view to represent the processes and activities of the enterprise; information view to represent the enterprise information used and obtained during the operation; resource view to represent the enterprise assets needed for carrying out the enterprise operations; and organization view to represent the organization, organizational relationships and the decision-making responsibilities in the enterprise operation." The standard is based on CIMOSA and GERAM [BN94]. Chapter 6 of the standard defines requirements on models and modelling method.

Furthermore, "Additional views for particular user concerns can be generated but these additional views are not part of this International Standard. Possible additional views are identified in ISO 15704 [ISO00b]."

### 2.2.3. Modeling Methods and Modeling Languages

No complete enterprise modeling method currently exists and there is serious doubt that it will ever exist. The most of the methods will also hardly address all these principles described. There exist a wide range of model types which can be used to describe aspects of an enterprise [Ver96]. Vernadat is referring to descriptive which are generally used for communication and common understanding for people in an enterprise, because of their informal, easy-to-learn syntax and formalism. Usually this type of models uses boxes, circles and arrows. Typical examples are IDEF [MM98], [MCFKP+95]), BPMN [BPMI03][OMG09], EPC [Sch93] [Kin04] and UML notations. And there exist more technical models with more formal description techniques such as Petri Nets [GAJV08].

Other researchers are looking for similar criteria's to categorize modelling languages i.e. Jablonski and Bussler [JB96], Zur Mühlen and Becker [zMB99] or Eder and Gruber [EG02].

Table 3 is summarizing modeling notations found in existing states. The following states have been considered: [ATHEN03], [BHABT+04], [Lan05], [UEML03]. These states will be introduced very briefly about their context and objectives.

**EU funded project: ATHENA State of the Art DA 1.1.1.Nb [ATHEN03]**

The ATHENA Interoperability Framework (AIF) provides a compound framework and associated reference architecture for capturing the research elements and solutions to interoperability issues that address the problem in a holistic way by inter-relating relevant information from different perspectives of the enterprise. The work is partly funded by the European Commission through the ATHENA IP (Advanced Technologies for interoperability of Heterogeneous Enterprise Networks and their Applications Integrated Project) (IST-507849). The deliverable which has been analyzed is about State of the Art in Enterprise Modeling Techniques and Technologies to Support Enterprise Interoperability.

**EU funded project INTEROP [BHABT+04]: "Deliverable D9.1: State-of-the-art for interoperability architecture approaches"**

This document analysed is titled "State-of-the art for Interoperability architecture approaches" with a focus on "Model-driven and dynamic, federated enterprise interoperability

architectures and interoperability for non-functional aspects". The aim is to provide a foundation for further analysis and work in the context of defining solution approaches and research issues related to the roadmap for interoperability related to Architecture&Platforms. Specifically, the analysed deliverable is stating modeling notations, modeling tools, modeling concepts, web-service based business processes and workflows.

**Lanckhorst [Lan05] "Enterprise Architecture at Work"**

This book is first giving a state of the art on EA Frameworks and modelling languages to then propose ArchiMate, which is an EA description language and EA Method. This has been created as an outcome from a project "Archimate", a Dutch research initiative that provides concepts and techniques to support enterprise architects in the visualisation, communication, and analysis of integrated architectures. The project consortium was consisting of Telematica Instituut, ABN Amro and many others.

**UEML (Unified Enterprise Modeling Language) [UEML03]: "D.1.1. Enterprise Modelling: State of the Art"**

The UEML project was set up in an attempt to contribute to the solving of the problems of multiple Enterprise Modelling Languages (EML). It is an IST Thematic Network funded by the European Commission in the Sixth Framework Program with the objective to create a European consensus on a common EML and to facilitate interoperability in the frame of on-going standardisation efforts in this domain. The state of the art is about enterprise modelling focusing on: Enterprise Modelling Languages (EMLs), Enterprise Engineering Tools (EETs) and Enterprise Modelling Methods (EMMs). This terminology has been re-used in table 3 to distinguish between these three categories. UEML has re-used definitions introduced by GERAM [BN94].

**Definition 24: Enterprise Modelling Language (EML) [BN94]**

> *"EML defines the generic modelling constructs for enterprise modelling adapted to the needs of people creating and using enterprise models. In particular enterprise modelling languages will provide construct to describe and model human roles, operational processes and their functional contents as well as the supporting information, office and production technologies."[BN94]*

**Definition 25: Enterprise Modelling Tool (EMT) [BN94]**

> *"EMT supports the processes of enterprise engineering and integration by implementing an enterprise engineering method and supporting modelling languages. Engineering tools should provide for analysis, design and use of enterprise models." [BN94]*

**Definition 26: Enterprise Modelling Method (EMM) [BN94]**

> *"EMM describes the processes of enterprise engineering and integration. An enterprise engineering method may be expressed in the form of a process model or*

*structured procedure with detailed instructions for each enterprise engineering and integration activity." [BN94]*

**Definition 27: Enterprise Meta-Modelling Languages (EMML) [BN94]**

*"EMMLs are languages that are used to describe enterprise modelling languages (their concepts, syntaxes and semantics), and to describe enterprise modelling methods." [BN94]*

Metamodels can help in the selection of application specific modelling language [zMue99] or supporting automation of processes in workflow [zMR99] [zMue02] [JBS97]. We will not focus on EMMLs and exclude this from the list in table 3.

A comparative study by zur Muehlen and Becker [zMB99] is today outdated as standards, methods and modelling languages are evolving very quickly. It will be more useful to concentrate on EMLs as they will be linked to available SOA methods in the next chapter. Finally, Van der Aalst, ter Hofstede and Weske [vdAtHW03] state that "all attempts to give an exhaustive overview over all methods, modeling languages and model types is predicted to fail." Hence, we will not claim to gather an exhaustive list, but a rather complete list, allowing preparing the ground for the picking of some notation candidates being suited to be used for the presented scope.

The presented list is a summary of existing state-of-the art of EMT, EML and EMM. They have been classified in alphabetical order including following information fields:

- Name
- Long Name
- Developer/Organization
- Year of Development
- Enterprise Modelling Tools (EMT), Enterprise Modelling Methods (EMM), Enterprise Modelling Languages (EML)
- *Popularity in SOA & BPM conference articles*
- *Links between modelling notations*
- *Standard Organization Support*

The last 3 criteria's will be used to get an indication on notations which are potentially suited as modelling notations in this thesis context. Therefore, the 3 criteria's need to be explained in more detail:

*The popularity in SOA & BPM conference articles* is important to identify the notations playing a role in the academic world fitting within the subject of process-oriented and model-driven SOA implementations. We can assume that published articles went through thorough evaluation process by specialists. Therefore, these articles using specific notations in that context can be considered as a reliable indicator. The main conferences on processes have been chosen in the period 2008-2012. Therefore, the Business Process Management (BPM2008 to BPM2012) conference articles (details table 5) and relevant IEEE conference papers (details table 6) have been screened and notation citations extracted. This is not claiming exhaustivity on BPM and SOA conferences, but is more intended to provide illustrative character. The details will be explained after table 3 in table 4.

The links *between modelling notations* is an important and neutral indicator which is important for a model-driven and process-oriented approach. Similar to the above mentioned criteria, published papers are used to rate if a notation can be linked to other notations situating on the same/different level of abstractions.

The *Standards Organization Support* criterion has the objective to provide some information on industry acceptance and utilization of standards. Furthermore, it can be considered that a well maintained standard notation achieves a level of quality and formalism such as meta-model, tool adaptation etc.

**Table 3: Overview on Enterprise Modeling Languages**

| **Descriptive Modeling Languages** | **Long Name** | **Developer / Organization** | **Year of Development** | **Tool (EMT) Method (EMM) Langu-age (EML)** | **Relevance in SOA & BPM conference articles (only for EML)** | **Links between modelling notations (only for EML)** | **Standard Organization Support (only for EML)** |
|---|---|---|---|---|---|---|---|
| ARIS [Sch93] | Architecture of Integrated Information Systems | Prof. Dr. A.W. Scheer | 1993 | EMT, EMM | | | |
| ArchiMate [Lan05] | | Marc Lanckhorst et al, ArchiMate project: Telematica Institute, ABN Amro etc. | 2005 | EMM, EML | | | |
| BMM [OMG10] | Business Motivation Model | Business Rules Group and adapted by OMG | 2008 | EMM, EML | | | X |
| BSC [KN92] | Balanced Scorecard | Kaplan, Norton | 1992 | | | | |
| BPEL [OASIS07] | Business Process Execution Language | IBM, BEA, Microsoft. Now OASIS | Creation 2002 by IBM, BEA, Microsoft, Now OASIS as from 2004 (WS-BPEL). Last release: 12.04.2007 | EML | X | X | X |
| BPDM [BPDM08] | Business Process Definition Metamodel | Open Management Group, OMG | Spec Version 1.0. published 3.11.2008 http://www.omg.org/spec/BPDM/1.0/volume1/PDF/ | EML | | | X |
| BPML [Ark02] | Business Process Modeling Language | BPMI, OMG | 2001 first draft by BPMI, Latest version supported by OMG 2002. Not | EML | | | X |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | supported any more | | | | |
| BPMN [OMG09] | Business Process Modeling Notation | BPMI, OMG | 2002 Steven White (IBM), adopted standard by OMG 2006 | EML | X | X | X |
| BOP | Business Operational Space | Computas AS | 2003 | EMM, EML | | | |
| CIMOSA [Ver92] | | ESPRIT consortium AMICE (>30 companies) | 1993 | EMM, EML, | | | |
| CORBA IDL | Common Object Request Broker Interface Definition Language | Object Management Group (OMG), | n.a. | EML | | | X |
| ebXML (corresponds to ISO 15000, includes XML based languages e.g. WSDL, SOAP, UDDI etc.) http://www.ebxml.org | Electronic Business using eXtensible Markup Language | joint initiative between the United Nations Centre for Trade facilitation and Electronic Business (UN/CEFACT) and Organization for the Advancement of Structured Information Standards (OASIS). | 1999 | EML | | | X |
| E3value Model [GP07] | | Jaap Gordijn | 2001 | EML, EMM, | | | |
| EEML [JC99] | Extended Enterprise Modelling Language | Eu-funded Project EXTERNAL, IST-1999-10091 | 1999 | EML | | | |
| EKS | Enterprise Knowledge Spaces | Lillehagen, Krogstie | n.a. | EML, EMM, | | | |
| EPC [Sch93] | Event-Driven-Process Chain | IDS Scheer | 1992 | EMM, EML | X | X | X |
| EDOC [OMG05] | UML Profile for enterprise distributed object computing) | OMG (Object Management Group) | 2005 | | | | X |
| FRAG [Zdu05] | | Uwe Zdun | 2005 | EML | | | |
| GRAI/GIM [VCZD91] | Graphes à Résultats et | Graisoft, Prof. Doumeingts, | 1984, GIM extension | EMT, EMM, | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Activités Interreliés | University of Bordeaux, France | 1992 | EML | | | |
| IDEF [MCFKP+95] | Integrated Computer-Aided Manufacturing Definition | US Airforce Program for Integrated Computer Aided Manufacturing (ICAM) | 1993 | EMM, EML | | | X |
| I* [YM93] | I-Star, "Eye-Star" | Yu and Mylopoulos | 1993 | EML | | | |
| IEM / MO2GO [SMJ96] | Integrated Enterprise Modelling | Fraunhofer Institute | 1996 | EMT, EMM, EML | | | |
| jPDL | Java Process Definition Language | Red Hat, JBOSS | n.a. | EML | | | |
| MEMO | Multi Perspective Enterprise Modeling | University of Koblenz (Germany), Prof Dr. Ulrich Frank | 1994 | EMT, EMM, EML | | | |
| METIS Enterprise | | Computas As | Release 3.4.7. June 2004 | EMT, EMM, | | | |
| MOF [JBR99] | Meta Object Facility | OMG (Object Management Group) | 2001/2002 | EMM, EML | | | X |
| NEML | Networked Enterprise Modelling Language | Spin Off of Testbed | n.a. | EML | | | |
| Petri Nets | | Carl Adam Petrie | 1962 | EML | X | X | |
| PIM4SOA [BL06] | Platform-independent model for service-oriented architecture | Benguria, Larrucea et al EU-funded ATHENA Project, European Software Institute (ESI) Spain, DFKI GmbH, Germany, SINTEF ICT, Norway | 2006 Status: Prototype | EMT, EMM, EML | | | |
| PIF http://ccs.mit.edu/pif1.html | Process Interchange Format | PIF Working Group | 1993 | EML | | | |
| PSL CORE, http://www.mel.nist.gov/psl/index.html | Process Specification Language | NIST: National Institute of Standards and Technology, USA | 1996 | EML | | | |
| SADT | Structured Analysis & Design | SofTech | 1977 | EMM, EML, | | | |
| SoaMl | Soa Modeling Language | OMG (Object Management Group) | 2009 | EML, | X | X | X |
| Testbed | | Telematica Institute, | 2004 | EMT, EMM, | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | BizzDesign | | EML | | | |
| UEML | Unified Enterprise Modelling Language | Research Initiatives started by ICEIMPT / NIST (1997), UEML Thematic Network project 2002-2003 | 1997, new work EU FP6 project 2002-2003 | EML, | | | |
| UML (and its profiles) | Unified Modeling Language | Object Management Group (OMG), Developers: Booch G., Jacobson I., Rumbaugh, J. | UML 1x:1997 UML 2.2. 2007 | EMM, EML | X | X | X |
| Value Chain [Por85] | | Michael Porter | 1985 | EMM | | | |
| WSDL [W3C01] | Web Service Description Language | W3C | 2001 | EML | X | X | X |
| WPDL [WFMC94] | Workflow Process Definition Language | Workflow Management Coalition (WfMC) | 1998 | EML | | | |
| XPDL [WFMC02] | XML Process Definition Language | Workflow Management Coalition (WfMC) | 1993,2002, Version 2.0 since 2005 | EML | | | |
| YAWL [vdAtH05] | Yet Another Workflow Language | Workflow Management Coalition (WfMC), van der Aalst/Ter Hofstede | First version:1999, 2004 integration into JBOSS | EML | X | | X |

Generally, the different state-of-the art deliverables from important EU-funded projects e.g. INTEROP, ATHENA, UEML are unfortunately not exhaustive because of rapid changes in model language evolution. To underline that, the following list gives an overview, in which deliverable, book or paper the enterprise modelling languages are explained.

Table 4 is indicating the sources, where information about the modelling languages can be found and which states have considered the review of these standards.

**Table 4: Summary on state of the art modelling languages**

| Modeling Languages | EU funded project: ATHENA State of the Art DA 1.1.1.Nb [ATHEN03] | EU funded project: INTEROP Deliverable D9.1: 'State-of-the-art for interoperability architecture approaches' [BHABT+04] | EU funded project UEML: D.1.1. Enterprise Modelling: State of the Art, UEML [UEML03] | Lanckhorst [Lan05] |
|---|---|---|---|---|
| ARIS | X | X | X | X |
| ArchiMate | | X | | X |
| BMM | | | | |
| BSC | | | | X |
| BPEL | X | X | | |
| BPDM | | | | |
| BPML | | X | X | |

| | | | | |
|---|---|---|---|---|
| BPMN | | X | | X |
| BOP | | | X | |
| CIMOSA | X | | X | |
| CORBA IDL | | | | |
| ebXML | X | X | | |
| E3Value Model | | X | | |
| EEML | | | X | |
| EKS | | | | |
| EPC | | X | | |
| EDOC | X | | X | |
| FRAG | | | | |
| GRAI/GIM | X | | X | |
| IDEF | X | X | X | X |
| IEM / MO2GO | X | | X | |
| jPDL | | X | | |
| I* | | | | |
| MEMO | | X | | |
| METIS Enterp | X | | X | |
| MOF | | | | |
| NEML | X | X | | |
| Petri Nets | X | | X | |
| PIM4SOA | | | | |
| PIF | X | X | | |
| PSL CORE | X | | X | |
| SADT | | | | |
| SoaML | | | | |
| Testbed | | | | X |
| UEML | X | | X | |
| UML | X | X | X | X |
| Value Chain | | X | | |
| WSDL | | X | | |
| WPDL | X | X | X | |
| XPDL | X | X | X | |
| YAWL | | X | | |

The following chapters will classify each notation on a specific level of abstraction. The execution code languages (XML) are not modelling languages per se, but are helpful for the execution of web-services in an orchestration language such as BPEL.

All notations that are not used anymore (BPML) or replaced by other notations are excluded.

For the structure of notations, we will use three different criterias:

1.) Suited Modelling Notations from practitioners feedback (by survey section 3.3.3.1., figure 34).
2.) Scientific popularity of notations (accepted articles from BPM and IEEE conference papers).
3.) Expressiveness of notations related to specific abstraction levels (as introduced in section 2.1.3.).

**1.) Suited Modelling Notations from practitioners feedback by survey (section 3.3.3.1., figure 34)**

The following table is the response from the exploratory feedback from practitioners on suited SOA modelling notations. The alphabetical list with notations had to be rated if the notation was A.) not known, B.) Known, C.) Known, used and meeting expectations or D.) Known, used and not meeting expectations. The details on survey design, method participants etc. are detailed in section 3.3.1.:

**Table 5: Suited Modelling Notation for SOA**

| Notation | Not known | Known | Known,used, meeting expectations | Known, used, **not** meeting expectations |
|---|---|---|---|---|
| BPEL | 9,26% | 66,67% | 22,22% | 1,85% |
| UML | 9,26% | 44,44% | 40,74% | 5,56% |
| BPMN | 20,37% | 51,86% | 27,78% | 0,00% |
| Value Chain | 25,93% | 46,30% | 25,93% | 1,85% |
| WSDL | 35,19% | 27,78% | 35,19% | 1,85% |
| BSC | 51,85% | 37,04% | 9,26% | 1,85% |
| EPC | 53,70% | 18,52% | 27,78% | 0,00% |
| IDEF | 55,56% | 35,19% | 5,56% | 3,70% |
| CORBA IDL | 68,52% | 27,78% | 3,70% | 0,00% |
| ebXML | 68,52% | 25,93% | 3,70% | 1,85% |
| WPDL | 70,37% | 25,93% | 3,70% | 0,00% |
| XPDL | 72,22% | 16,67% | 11,11% | 0,00% |
| Petri Nets | 77,78% | 16,67% | 5,56% | 0,00% |
| e3 Value | 81,48% | 12,96% | 1,85% | 3,70% |

The table filters the notations on top, which are the mostly known (low percentage on “not known”.

**2.) Scientific popularity of notations**

This selection has been done based on citation of these notations in accepted papers of BPM conferences 2008 to 2012. Only one occurrence per modelling notation per paper was possible.

Additionally to the BPM conference papers, a broader request on modelling notations within all IEEE conferences has been launched on IEEE Explore [IEEE12]. The search has been done for the time range between 2007 and 2012. The notation has been put in the SOA context by requesting: “Modeling notation” AND “SOA” - as a full-text and metadata search:

**Table 6: Citation of Notations Comparision Academic Conferences**

| Notation | BPM Conferences Citations | IEEE Conferences Citations | BPM Rank | IEEE Conferences Rank | Average Rank Score ((BPM+IEEE)/2) |
|---|---|---|---|---|---|
| BPMN | 82 | 170 | 1 | 7 | 4 |
| BPEL | 54 | 663 | 2 | 4 | 3 |
| EPC | 51 | 693 | 3 | 3 | 3 |
| Petri Nets | 42 | 2257 | 4 | 1 | 2,5 |
| UML Activity Diagram | 30 | 225 | 5 | 6 | 5,5 |
| YAWL | 18 | 13 | 6 | 11 | 8,5 |
| WSDL | 16 | 427 | 7 | 5 | 6 |
| IDEF | 6 | 27 | 8 | 9 | 8,5 |
| Open Work Flow Nets | 6 | 0 | 8 | 12 | 10 |
| BSC Model | 2 | 99 | 9 | 8 | 8,5 |
| E3 Value Model | 2 | 5 | 9 | 12 | 10,5 |
| Value Chain Model | 2 | 793 | 9 | 2 | 5,5 |
| KAOS Model | 1 | 22 | 10 | 10 | 10 |
| I* Model | 1 | 0 | 10 | 12 | 11 |
| Tropos Goal Risk Model | 1 | 0 | 10 | 12 | 11 |

The citations count of notations in academic papers give an indication of academic popularity of notations. The ranking indication can be used to identify notations which might be more in the focus of interest than others. Again, notations on strategic level seem generally to be less cited in conference and workshop papers.

## 3.) Expressiveness of notations

One quality property of modelling notations as proposed by Hommes and Van Reijswoud [HR00] based on the FRISCO report [FHLNH+98] is expressiveness. By expressiveness we understand

**Definition 28: Model Expressiveness [FHLNH+98]**

> *"the degree to which a given modelling technique is capable of denoting the models of any number and kinds of application domains." [FHLNH+98]*

Formalized meta-models and also the ability to transform from one denotation into another are measures for high expressiveness [HR00].

The following table will indicate with a rather basic scale (high-medium-low) the expressiveness of notations related to the MDA abstraction levels introduced in section 2.1.3., where research such as from [NK06] [RRIG09] is used as input:

**Table 7: Expressiveness of Notations related to the MDA abstraction level**

| Notation | MDA Abstraction Level | Expressiveness (High-Medium-Low) |
|---|---|---|
| BMM | Strategy | Medium |
| Value Chain | Strategy | Low |
| BSC | Strategy | Low |
| I* | Strategy | Low |
| Strategic Planning | Strategy | Low |
| e3 Value | Strategy | Medium |
| BPML | CIM | Medium |
| EPC | CIM | High |
| IDEF | CIM | Medium |
| BPEL | PIM | Medium |
| UML | PIM | High |
| BPMN | PIM | High |
| PETRI NETS | PIM | Medium |
| WSDL | PSM | Low |
| CORBA IDL | PSM | Low |
| ebXML | PSM | Low |
| WPDL | PSM | Low |
| XPDL | PSM | Low |

The presented tables 5 to 7 will help through the next sections to concentrate on the notations seeming the most suitable for a top-down modelling approch using popular notation in practice and academia with high expressiveness of notations on their respective level. On the strategic level, most notations are de facto not known and also not very expressive. However, these notations are in our scope important as we motivate an approach where also strategy should be formalized in models. This will be further detailed in section 2.2.5.

## 2.2.4. Business Process Management as Framework for Modeling

When entering the field of business process modeling, an overwhelming number of tools and modeling languages are available. Often these languages and tools have very little in common. In most of the cases, the conceptual domains that are covered differ from language to language. Some emphasize elements of workflow in the models, others concentrate on quantitative analysis and others try to integrate business processes and supporting information technology. Moreover, software tools are an important success factor for a language; some of the most popular languages e.g. ARIS [Sche93] are proprietary to a specific tool. It is clear that none of them has succeeded to become "the standard language" [BHABT+04].

However, modelling needs to be seen in a broader context of business process management (BPM). BPM, with its critical success factors [BGR07] is the discipline of managing processes with the help of models for a specific business objective. The business objectives (concerns) can be “documentation”, “certification”, “improvement”, “risk-and compliance”, “application development” and many others. Depending on the objectives, the viewpoint and focus of what is relevant will change. For instance an improvement objective will more focus on cost, time and quality using eventually activity-based-costing method than risk-and

compliance, which is more focussing on control activities and the related test to ensure effectiveness of controls. The discipline of BPM is supporting the achievement of these expected results. The relevance for this thesis has to be seen in the application of process models on the viewpoint of modelling, but also on the viewpoint on knowledge how to model and what to model to cope with the concern of SOA Method.

In figure 7, Karagiannis et al [KJS96] have defined different processes exemplarily for BPM. The content of the processes has no relevance for the SOA Method, but it illustrates the different levels of abstractions. The ability to structure and perform these processes is enhanced by tools represented as a list on the right side of the figure 7 [KJS96]. Additionally, 4 levels are used, which are similar to the earlier introduced abstraction layers of CIM added by an additional level which is strategy.

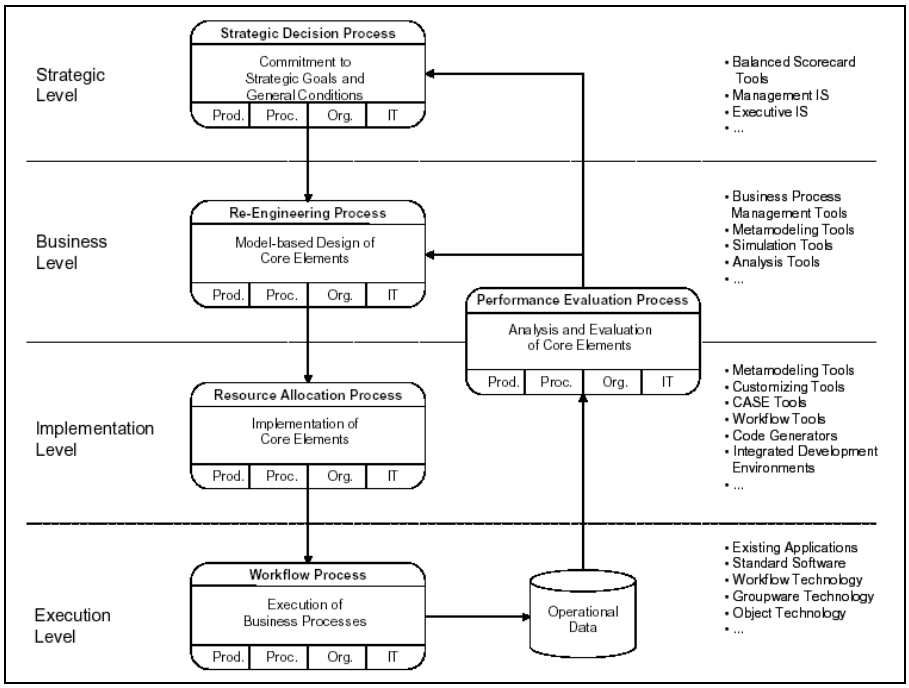



**Figure 7: The Business Process Management System Paradigm**

A myriad of other examples for BPM systems and components could be found such as in [JK04] or [BR10b] with the same content but different words or ontologies. Important in this context is to recognize the different level of abstractions with strategic level, CIM, PIM and PSM. Available modelling notations can be linked to these levels to formalize the necessary modelling content following the specific objectives. The next sections will illustrate this in detail.

## 2.2.5. Business Strategy Concepts

This section will quickly introduce business strategy concepts and different approaches. These Strategy definition approaches will be described and finally strategy definition elements in SOA discussed. The reason is that SOA requires a more business value view than technical viewpoints as declared in the SOA Manifesto [SOA09] where a working group developed a set of objectives and guiding principles aiming to provide a better understanding of SOA. Key prioritizations were “Business Value over Technical Strategy” and Strategic Goals over

project-specific benefits". Therefore, strategic concepts have to be investigated for SOA Methods in relationship with modeling.

**Definition 29: Strategy [Por96] [Dru94] [MAL98]**

> *"Strategy is the creation of a unique and valuable position, involving a different set of activities."* [Por96]

and

> *"Strategy is about knowing where your company is today, where you want to take it, and how you are going to get there."* [Dru94]

and

> *"Strategy is aiming to set the direction, to focus effort, to define organization and to provide consistency."* [MAL98]

Business strategy concepts are a very complex and vast area related to different views. The concept of views has been already introduced in section 2.1. Mintzberg, Ahlstrand & Lampe in their book "Strategy Safary" [MAL98] describe 10 different views on the strategy process:

1. The design school: Strategy formation as a process of conception
2. The planning School: Strategy formation as a formal process
3. The positioning school: Strategy formation as an analytical process
4. The entrepreneurial School: Strategy formation as a visionary process
5. The cognitive school: Strategy formation as a mental process
6. The learning school: Strategy formation as an emergent process
7. The power school: Strategy formation as a process of negotiation
8. The cultural school: Strategy formation as a collective process
9. The environment School: Strategy formation as a reactive process
10. The configuration School: Strategy formation as a process of transformation

To comply with the research subject, the strategies need to be translated into a model. This means that we need to find a description language and a tool with the ability to do so. Known methods for strategy implementation and support by a tool are the "Balanced Scorecard" and "Value Chain".

Kaplan and Norton [KN92] [KN93] introduced the BSC as a management system that helps an enterprise to clarify and implement its vision and strategy. The BSC therefore suggest to view an enterprise from four perspectives (Financial, Customer, Process and Learning and Growth) decomposed into a three-layered structure: 1. Mission (e.g. become the customers' preferred supplier), 2.Objectives (e.g., to provide the customers with innovative products) and 3. Measures (e.g., % of turnover generated by new and innovative products).

The original concept of Value Chain was created by Porter [Por85]. The chain consists of a series of activities that create and build value. They culminate in the total value delivered by an organization. The concept of "margin" is equal to added value. The organization is split into "primary activities" and "support activities". These functions can be linked to one another in the form of a sequence of functions and thus form a value-added chain. The value

chain is a systematic approach to examining the development of competitive advantage. The drill-down of each business function is necessary to show how the functions are performed.

Giannoulis et al. [GPZ10] have analysed the formalization of strategy maps [KN04a] [KN04b] and Balanced Scorecards [KN92] [KN93] with the objective to formalize strategy maps in the form of a meta-model, usage scenarios and constraints to achieve a unified language/ontology for business strategy modelling:

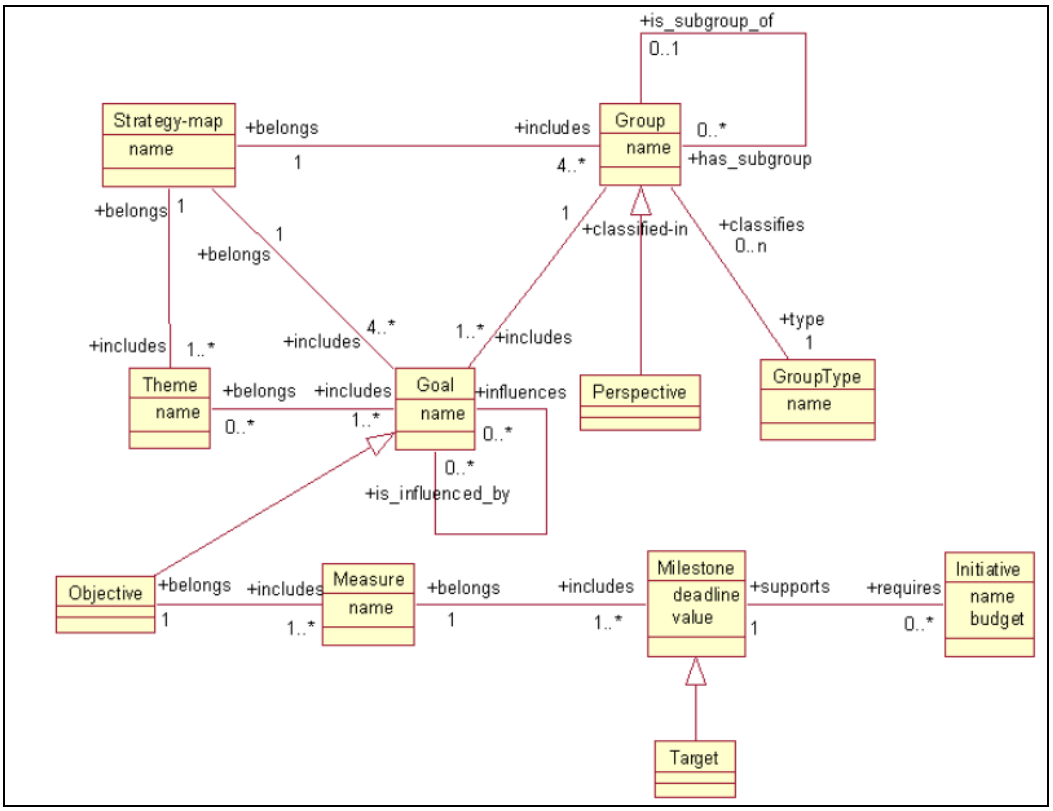



**Figure 8: The Value Configuration Meta-Model [GPZ10]**

The meta model consists of classes and cardinality constraints for the relations between classes. Following to [NEKZ+05], a process is executed to satisfy a goal (goal class). This is where the link is to next deeper levels of process landscapes expressed by value chains. This is also a future work area of [NEKZ+05] to provide enriched meta-models with the objective to transform business strategies to lower-level model e.g. business process models.

Following to Rigby [Rig07], the most used approach is strategic planning. After the first place in 2005, strategic planning also ranked in 2006 at the top-level [RB07]:

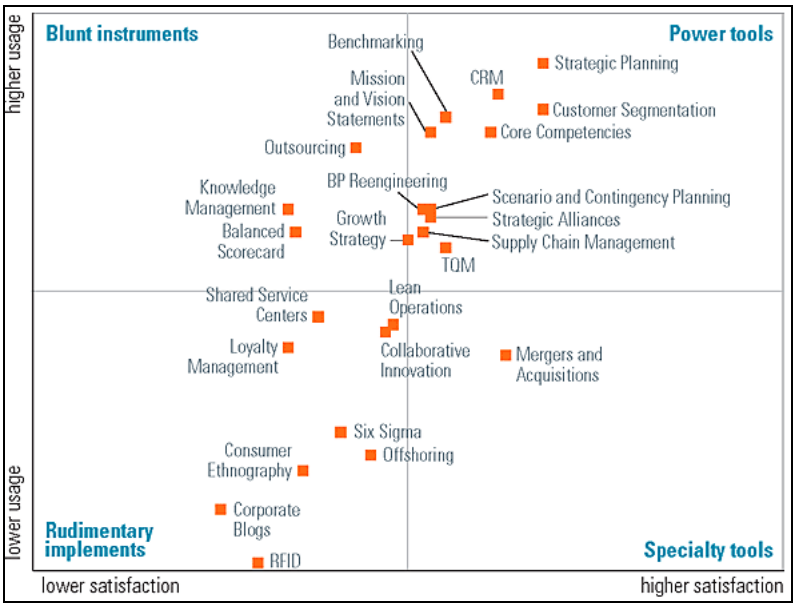



**Table 8: Positioning of Strategic Tools**

So far, no modelling notation for strategic planning has been discovered. The key criteria for a consistent and integrated approach and method for the implementation of SOA is therefore the ability of the tool to link the strategic model to the processes or process models.

Recently, some work [DP07] has shown the relationship between business model, business process model, business goals and business requirements:

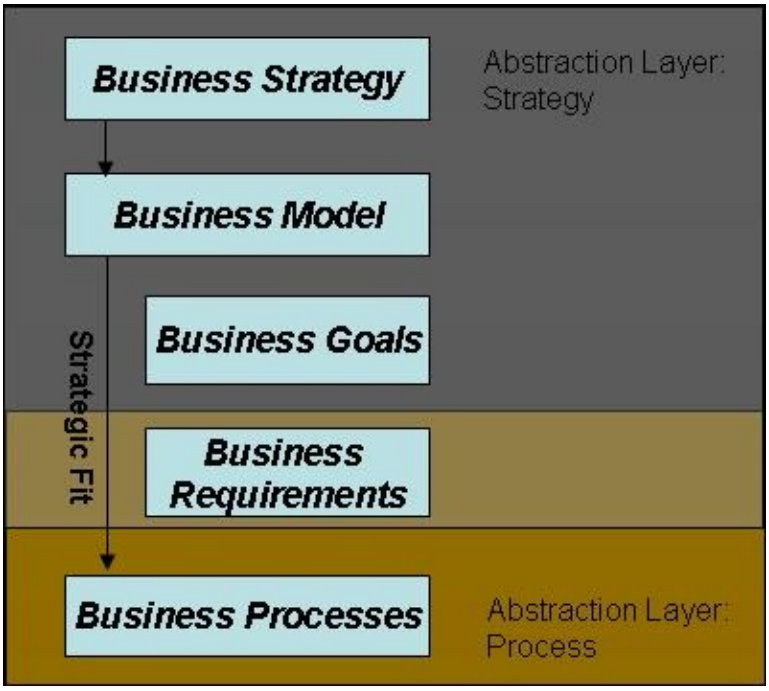



**Figure 9: Refined Abstraction layer Strategy adapted from [DP07]**

This figure above is a finer grained overview of the strategy layer introduced in figure 6.
The business Strategy can be derived from the vision and mission of an organization. Next, business model and business goals can be positioned on the strategy level. The business requirement is the link to business processes. “Strategic fit” can be checked between business model and business processes.

The Strategic Alignment Model (SAM) developed by Henderson and Venkatraman [HV93] is considered as the key reference alignment model:

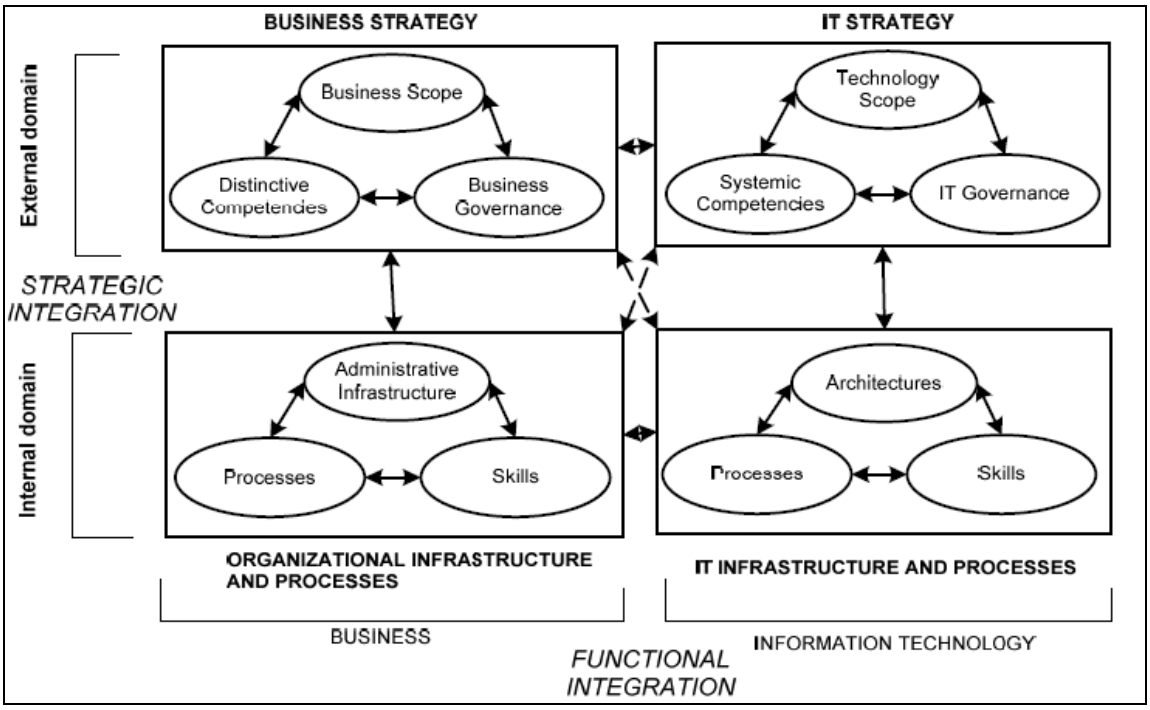



**Figure 10: The Strategic Alignment Model (SAM) [HV93]**

This alignment model includes four main components to consider for alignment, i.e. business strategy, IT strategy, organizational infrastructure and processes and IT infrastructure and processes. Next, it also specifies two types of integration which is a.) the strategic integration between the business strategy and the IT strategy in the context of the external domain and b.) the functional integration between the business organizational infrastructure and processes and the IT infrastructures and processes in the context of the internal domain.

In the context of SOA implementation method, the strategic direction (Business Strategy and IT Strategy) has to be aligned with the processes and architecture defined as the internal domains. Following to Prado [Pra09], "alignment is a continual adjustment process of conscious and coherent interrelation of all business and IT components and personnel in order to contribute appropriately and quickly to the business goals and needs over time"

Next, we will explore another area of the strategy layer, which is about the business model:

**Definition 30: Business Model [OP10]**

> *"A business model is a conceptual tool that contains a set of elements and their relationships and allows expressing the business logic of a specific firm. It is a description of the value a company offers to one or several segments of customers and of the architecture of the firm and its network of partners for creating, marketing, and delivering this value and relationship capital, to generate profitable and sustainable revenue streams."* [OP10]

4 pillars composing nine building blocks have been identified [DP07] in this Business Model Ontology (BMO):

**Table 9: Nine Business Model Building Blocks**

| Pillar | Business Model Building Block | Description |
|---|---|---|
| Product | Value Proposition | Gives an overall view of a company's bundle of products and services. |
| Customer Interface | Target Customer | Describes the segments of customers a company wants to offer value to. |
| | Distribution Channel | Describes the various means of the company to get in touch with its customers. |
| | Relationship | Explains the kind of links a company establishes between itself and its different customer segments. |
| Infrastructure Management | Value Configuration | Describes the arrangement of activities and resources. |
| | Core Competency | Outlines the competencies necessary to execute the company's business model. |
| | Partner Network | Portrays the network of cooperative agreements with other companies necessary to efficiently offer and commercialize value. |
| Financial Aspects | Cost Structure | Sums up the monetary consequences of the means employed in the business model. |
| | Revenue Model | Describes the way a company makes money through a variety of revenue flows. |

The pillars of this BMO are partly similar to the "strategic objectives" defined and suggested by Kaplan & Norton in their Balanced Scorecard approach [Rig07].

The e3value business ontology was originally proposed to model the value networks of cooperating business partners [GAV00]. The ontology aims at identifying the exchanges of objects of economic value (value objects) between the involved actors in business collaboration. The e3value ontology provides a rich set of software tools to design and analyse value webs, including a graphical notation. It also provides a minimal set of concepts and relations, thus making it easier to be understood by all the involved stakeholders. Figure 11 is taken from a case study [GA03] about online news provisioning is illustrating the e3value ontology:

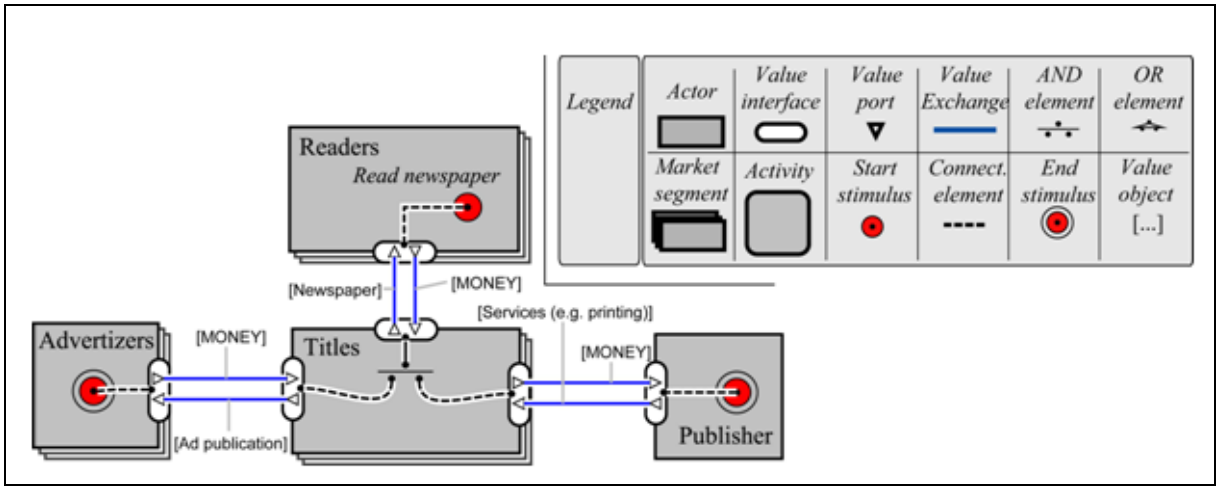



**Figure 11: e3value model for problem analysis**

This model notation has been initially developed for e-business scenarios. A considerable effort is done to expand this notation also to other use cases not directly linked to e-business. The promise of this notation and the research done in that area is about deriving from the business model notation (strategic) a process model notation (business process). There are also different researchers enhancing the e3value model by complementary views and notations e.g. the e3forces ontology is introducing constructs for representing and modelling

strategic motivations from environmental forces [GP07]. Pijpers, Gordijn and Akkermans [PGAa09], [PGAb09], [PGAc09], are exploring e3strategy and e3alignment to business and organizational aspects. The work is based on Porter's five forces [JS02], [Por80], [Por85]: bargaining power of suppliers, bargaining power of buyers, competitive rivalry among competitors, threat of new entrants and threat of substitutions.

An example [GP07] taken from the passenger aviation industry is the introduction of e-ticket system. The business model highly depends on this IT system providing significantly cost reduction per booking process, which can be used to reduce price to achieve more value for money and attract more customers. An e3forces model helps by determining where IT can create competitive advantage by providing a graphical overview of relationships with markets [GP07].

An i* model has been used in the approach of comparison/mapping/evolution between business models by the INTEROP project [YM93] to show the "Why" of a business model, whereas the value models focus on "What". Some of the requirements and goals of actors involved do apply to the characteristics of the value transfers; other goals can be derived from the business context and the objectives of the common value creation [Yu95]. YU and Mylopoulos developed I* for capturing this business context [YM96] based on goal-oriented techniques helping reasoning on the business of an organization and on its associated objectives. Various researcher teams [GPW06] [RGY05] are explaining the benefits of goal and value modelling to operationalize business strategy concepts. An easy to understand application of three actors (Seller, Warehouse and Buyer) is illustrated in figure 12 [DP07]:

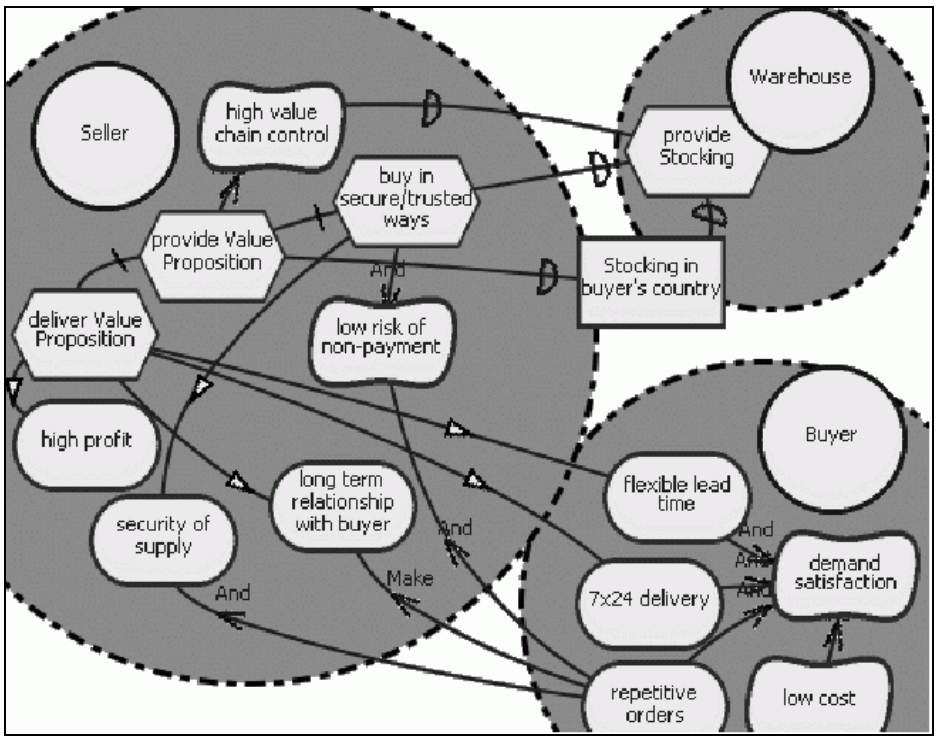



**Figure 12: I* illustration [DP07]**

The i* is therefore complementary to the value notation to express missing information on "why" of business model and giving details of business context. This allows representing strategic objectives in a similar way to BSC which uses "Strategic objectives" in their four perspectives. The objectives in both models correspond to each other. The i* notation was also part of a comparison analysis with state-of the-art notations for goal driven requirements

engineering [KL03].The view presented is different from this thesis, but i* (or tropos, a dialect of i*) has been evaluated as the only notation with formal representation and tool support. Other goal models or goal analysis techniques such as goal based workflow, cognitive task analysis, EKD, F3(OM), ISAC, SIBYL, the reasoning loop model, REMAP, KAOS, GEBRAM, Goal-scenario coupling, NFR Framework, GSN and GQM are evaluated and is a research topic for itself and ongoing.

Another model in the strategy abstraction layer is the OMGs Business Motivation Model (BMM) [OMG10]. BMM positions itself as a structure for "developing, communicating, and managing business plans in an organized manner." According to OMG, the BMM

- identifies factors that motivate the establishment of business plans.
- identifies and defines the elements of business plans.
- indicates how all these factors and elements inter-relate, and is furthermore providing governance and guidance by policies and business rules.

All the used terms in the BMM are described in detail and expressed through a meta-model and a fact based model.

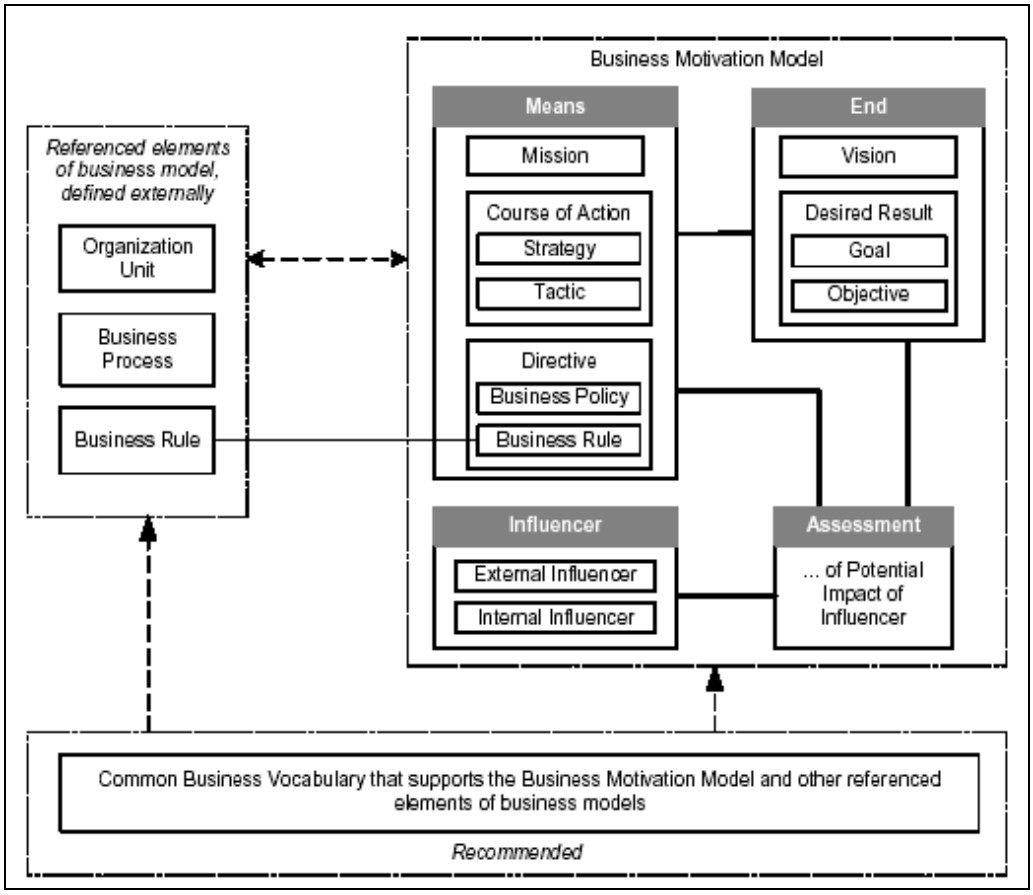



**Figure 13: Business Motivation Model OMG (BMM)**

BMM is also showing in their referenced elements box "Business Process". As notation, BMM in the specification document [OMG10] is referring to the BPMN standard through the externally defined element.

Figure 14 is also illustrating the BMM model, but Berkem [Ber08] enhanced the model with more information on the links in-between the concepts and also explaining the link to SOA application scenario.

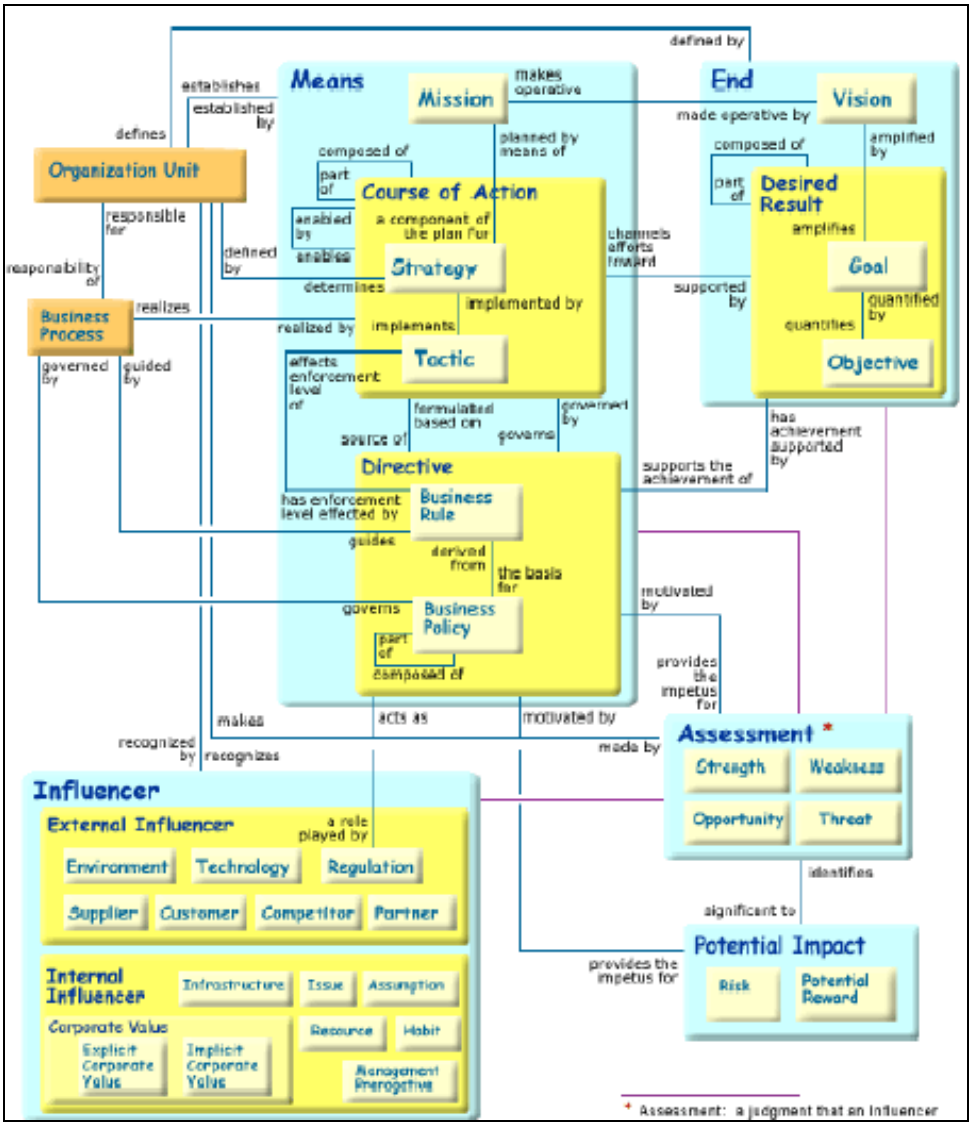



**Figure 14: Business Motivation Model enriched [Ber08]**

Berkem [Ber08] is arguing that the reason for performing processes always is starting on strategy level. Business processes realize then actions with the objective to fulfil strategic objectives and goals.

Some of the logic of business processes may be expressed in business rules. Business rules are derived from business policies.

Since much of the motivation for what an organization does is based on people in the organization deciding what is best for it, the organization should be able to say who decided, and on what assessments of what influences. In practice, real businesses do not have complete traceability of motivation. But, as and when they choose to move towards it, the BMM is a possibility to support it.

This can be used to make strategy and goals more visible. Tools exist (e.g. Select Business Modeler) in supporting the presented BMM and linking to process modelling notations.

The link of earlier mentioned strategy models to a SOA is the following: these types of models such as BSC, i*, e3forces and e3-value can be used to formulate strategy and business model into a graphical notation.

This section has provided some insight into what strategy is and which tools or methods can be found on this level of abstraction.

The next section will therefore focus on the interface between strategy models and process models.

## *2.3. Interfaces between Abstraction Layers*

The following sections will explore which could be candidate modelling notations on different levels of abstraction. In particular the positioning of notations and their possible transformation and mapping mechanisms are interesting.

### 2.3.1. Interface between Strategy layer and Process Layer

On the strategic level as presented earlier, different possibilities to represent strategy in models are existing:

- E3-value
- E3-forces
- i*
- Balanced Scorecard
- Strategic Planning
- BMM

All analysed SOA methods neglect this type of strategy modelling. Some are giving advice to include strategic objectives, but no SOA method includes one of the 6 mentioned notations.

E3value and e3-forces are quite close to each other, as e3forces has been developed based on e3-value. Both notations have their roots and basic idea from Porters work on strategy. The alignment of business strategy of an enterprise with the required information technology needed to enable the e-service in a networked value [PGA08] was analysed. Their approach is claiming to enhance other frameworks for Strategy-IT/IS alignments as described in [Bae92][HV93][LPB95]. One recent approach is proposing the following constellation setting [DG06] in figure 15:

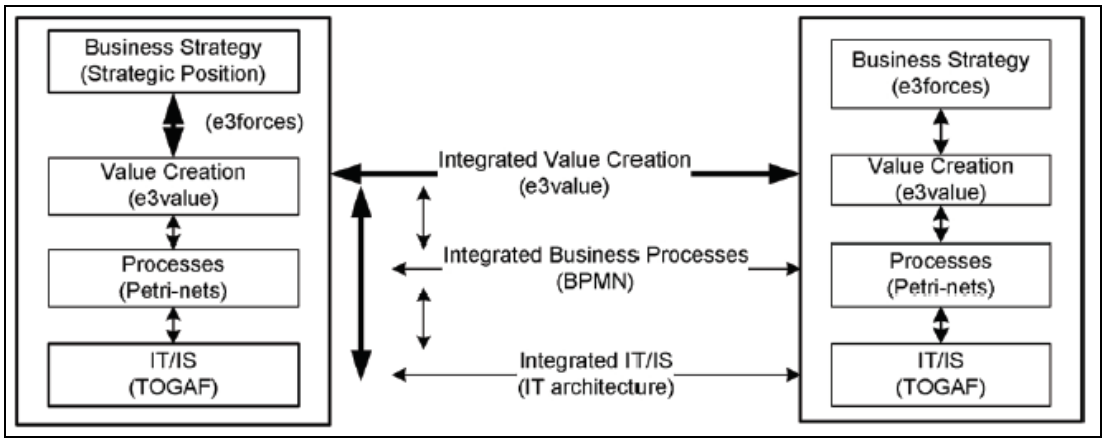



**Figure 15: Inter-organizational alignment model [DG06]**

We see an example independently from the SOA context of aligning modelling methods on different level of abstractions. E3value and e3forces modeling is used on the strategy level which is connected to the processes (petri-nets) and IT/IS using TOGAF as EA model. BPMN is used as interfacing notation between the petri-nets.

A semi-automatic translation from e3-value to BPMN has been proposed by Edirisuriya and Johannesson [EJ08] by exploring how a process model can be systematically derived from a business model. The paper presents an enhanced solution towards proposed e3transition model introduced in Pijpers and Gordijn [PG07]. A developed Activity Dependency Model (ADM) is used to interface e3-value on strategy abstraction layer and BPMN on process abstraction layer. The purpose of an ADM is to bridge the gap between business models and process models. An ADM provides more details than a business model and fewer details than a process model. It identifies and classifies the activities that are necessary to exchange resources, produce resources, deliver services, and the relations that exist among these activities. Transformation rules from e3value to ADM (6 rules) and between ADM and BPMN (9 rules) define the mapping from one model to another. Furthermore, four primitive value transfer process patterns are used to show exchange of business objects between provider and recipient. Future research will test the completeness and correctness of the mapping rules in other case studies. The BMM and Strategic Planning are more a set of guidelines and frameworks than real modelling notations. They might be helpful and act as a starting point to fill/construct e3-forces, i* or BSC models.

This link from strategy to process is very difficult to make and also mostly a manual process. For the decision model, it is therefore recommended to consider the following notations:

The e3-value and e3-force notation are already recognized notations by academia. A restriction might be the focus on business models based on web technology. Classical industries e.g. manufacturing, logistics, supply mgt. where the focus is not put on the web as sales channel could lead into issues in translating the business model into e3-notations. On the other hand, case studies on air carrier [PG07] or banking [KG07] have been conducted.

The translation from e3-value to BPMN is possible via ADM mapping rules. Strategic objectives should also be linked to processes. Therefore the I* or the BSC could be used. The content of BMM can be taken as input for the I* and/or the BSC.

### 2.3.2. Interface between Process layer and IT Layer

Abstract models and semi-formal notations are stepwise refined and linked to the next deeper layer. Business-oriented process models tend to be incomplete representations of processes according to implementation relevant details. For instance exception handling for unexpected events, such as special cases from a business point of view or technical failures, is often omitted [DvdA04]. Another path described by Stein et al. [Stei08] is starting with business requirements formalized in an EPC process model. An automatic translation into BPEL is demonstrated through an eGovernment case study. However, notations to formalize strategy and linked objectives and business requirements have been neglected. EPC is an excellent alternate solution to BPMN as standard transformation engines are able to automate from EPC to BPEL (ARIS SOA Architect) [Stei08].

The strategy can be formalized in different notations as presented. Strategy models and process models are two types of model in a chain of models used by enterprises to describe different aspects. Hence the strategic model provides a high level view of the activities taking place within and between actors, how value is generated and what strategic objectives are set. The process models are taking into consideration the given strategy models, but the presented notations will more focus on explaining operational details how business is executed. When entering the field of process layer, different modelling notations are available. We will first list the notations where a bridge from strategy can be made:

- Value Added Chain
- EPC
- BPMN
- UML (Activity Diagram)

No bridge identified for:

- Petri Net
- IDEF

The OMG Profile SoaML, which is a specification for the UML Profile and Metamodel for Services Version 1.0. beta [OMG09b] addresses a modelling solution focussing on web-services choreography, interfaces and interaction. A meta model (based on UML2) is proposed and integrating MDA principles. Following to OMG [OMG09b] “SoaML focuses on the basic service modeling concepts, and the intention is to use this as a foundation for further extensions both related to integration with other OMG meta models like BPDM and BPMN 2.0, as well as SBVR, OSM, ODM and others.” Chapter 9 of the beta version document is indicating a connection to the OMG BMM model. The motivation element (a Vision, Goal, Objective, Mission, Strategy, Tactic, Business Policy, Regulation, etc.) is linked to UML scenarios or UML activity diagram.

An already presented academic approach [ZD06] is taking care of translating process models into execution notations on the IT layer. In most of the cases BPMN and UML play a central role. The BPMN standard developed by BPMI [BPMI03] has been taken over by OMG [OMG06] in 2006. Initially OMG positioned BPMN as a “notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes. Thus, BPMN creates a standardized bridge for the gap between the

business process design and process implementation." In 2006 at an early stage, this was not completely true, but meanwhile the standard has much evolved and the latest version BPMN 2.0. [OMG11] became a real alternate choice to the business requirements modelling notations. Both can be used to translate into BPEL and WSDL. As UML has been created and used for software development, the notation is well suited to specify implementation details. The more promising notation with better connection to execution level with technical orchestration models is certainly BPMN with associated schema files of XMI, XSD, XSLT [OMG11]. No bridge has been identified for Petri Nets and IDEF.

### 2.3.3. Summary of notation capabilities

In order to outline the selected notations for a model-driven and process-oriented SOA implementation on the various levels, the table below is summarizing the notations. It is important to note that this is not an exhaustive and detailed abstract – to do this by also including details on strengths and weaknesses would be a topic of another thesis. The objective here is to resume the presented notations candidates found along the different levels of abstraction.

**Table 10: Notation Description Summary**

| Notation | Description |
|---|---|
| Strategic Planning Model | Strategy formulation and formalization with mission and vision statements, strategic objectives, action plans with budgets. |
| Balanced Scorecard Model | Strategic objectives based on cause-and-effect relationships, not only focussed on financial objectives but also on other dimensions e.g. learning/education (internal), processes, customers and finance. |
| E3forces | Strategy Modeling notation including environmental criteria's (based on Porters 5 forces) and based on environmental strategy school. |
| E3Value | Business Model Notation with focus on value exchange between actors. |
| I* | Is used to show in an explicit way the goals of an organization. A goal can have sub-goals and influence other goals. |
| Value Added Chain Diagram | Based on Porters Value Chain used for "big pictures" or functional processes / macro processes overview. |
| BPMN 2.0. | Business Process Notations (Business Requirements) issued by OMG, process capabilities, process choreography, business rules management. |
| EPC | Business Process notation (Business Requirements) ability to enhance process sequence by data, application, organization elements and rules. |
| UML Diagrams | Modelling notation (Requirement but also technical), linked to meta model MOF, ability to represent processes with a comprehensive set of profiles such as SoaML. |
| IDEF | Business process notation (Business Requirements) with IDEF0 for functional modelling, IDEF3 for workflow and IDEF1X for data modelling. |
| Petri Net | Business process notation (Business Requirements) with main focus on workflow (tokens) mainly used in the manufacturing industry. |
| BPEL (= BPEL4WS) | Coordinating the execution of business process including web services (call, loop, run, exceptions etc.) "orchestration" language. |
| WSDL | XML based language to describe web-services. |
| YAWL | De-facto standard for executable workflow models. |

## 2.4. Model Transformation

### 2.4.1. Model Transformation Mechanisms

When deciding for a specific notation and a path down through the different abstraction levels, an important criterion is the ability of notations to transform to the next deeper layer. We need to distinguish between "model translation" and "language translation". The first is about the definition of mappings between models in the same language and the second is about mappings between models in different languages [Ken02]. We will in this context only look at "language translation" as different languages are involved on the different layers of abstraction. We therefore refer to the following definition

**Definition 31: Model Transformation**

> *"The ability to transform a model based on a meta-model or pre-defined semantics into another model or code".*

We will still maintain the term "modeling language" as it is better understandable:

There are two basic types of modeling language transformations:

- model-to-model and
- model-to-code

A model-to-model transformation maps a model related to a given meta-model to another kind of model conforming to another meta-model. The result can also be similar without meta-model if transformation rules are applied. Furthermore, a manual, semi-automatic and automatic transformation needs to be considered. In the literature there are numerous code generation techniques such as templates and filtering, template and meta-model, inline generation, code weaving, etc. [VS06]. Model-to-code or a synonym for code generation produces executable code from a specific source model. For the automatic transformation generally, there are two schools of thought:

- Transformation based on pre-defined semantics and
- Transformation based on meta-models

Some tools on the market (open source vs proprietary) are able to support those two transformation types. The objective within this thesis is to provide a complete view on different approaches related to automation and specific types of models on each abstraction level.

The most research in this area is done by research teams with an informatics background. Business analysts usually design processes in high abstraction languages, such as BPMN, EPC, or UML Activity Diagram, and developers implement them using executable languages, such as BPEL/WSDL. An important issue that hinders the interoperability and the reusability of existing process models is the huge divergence of these modelling languages. This issue occurs because there is no explicit link between two modelling languages at the same or different abstraction levels. For instance, developers could not re-use or integrate the whole or part of a process described using BPEL in another process developed using BPMN or EPC,

and vice versa. The most popular solution for this issue is to define direct transformations based on pre-defined semantics between the different process modelling languages [MLZ05], [MZ05], [ZM05], [RM06].

A main view with the concern to show business rules is dedicated to the control flow within a process model. Even after more than ten years of standardization efforts [Hol04], the primary BPM languages are still heterogeneous in syntax and semantics. This problem mainly relates to two issues: Firstly, various BPM language concepts that need to be specified in terms of control flow [vdAtHK+03] and data flow [RtHE+05] have been identified, and most BPM languages introduce a different subset of these [MNN04]. Secondly, the paradigm for representing control flow used in the BPM languages is another source of heterogeneity. This issue has not been discussed in full depth so far, but it is of special importance when transformations between BPM languages need to be implemented. In essence, two control flow paradigms can be distinguished, graph- and block-oriented [MLZ06]:

**Graph-oriented BPM languages** specify control flow via arcs that represent the temporal and logical dependencies between nodes. A graph-oriented language may include different types of nodes. These node types may be different from language to language. Workflow nets [vdA97] distinguish places and transitions similar to Petri nets. EPCs [KNS92] include function, event, and connector node types. YAWL [vdAtH05] uses nodes that represent tasks and conditions. Similar to XPDL [WFMC02], these tasks may specify join and split rules.

**Block-oriented BPM languages** define control flow by nesting control primitives used to represent concurrency, alternatives, and loops. XLANG [Tha01] is an example of a pure block-oriented language. BPML [Ark02] and BPEL [ACDGK+03] are also block-oriented languages but they also include some graph-oriented concepts (i.e.links). In BPEL, the control primitives are called structured activities. Due to the widespread adoption of BPEL as a standard, we will stick to BPEL as an example of a block-oriented language.

Table 11 is summarizing the transformations following the degree of automation and abstraction level:

**Table 11: Model Transformation Overview**

| Degree of Automation / Abstraction Level | Automatic (pre-def. semantics & meta-models) | Semi-automatic & manual |
|---|---|---|
| Model-to-model | EPC2BPEL<br>BPMN2BPEL<br>UML2BPEL<br>BPEL2EPC | e3value2ADM<br>ADM2BPMN<br>EPC2BPMN<br>EPC2UML |
| Model-to-code | BPEL2WSDL<br>UML2WSDL | |

Transformation based on pre-defined semantics is done by using transformation strategies as explained in [MLZ06]: Element-Preservation (e.g. EPC2BPEL [ZM05], UML2BPEL [ZM05]), Structure Identification (e.g. BPMN2BPEL [Gar03]), Event-Condition-Action-Rules (e.g. BPMN2BPEL [OvdA+06] and Flattening (e.g. BPEL2EPC [ZM05]).

Furthermore, Mens et al. [MCvG05] define success criteria's, characteristics and quality requirements of model transformation.

The linked topic of interoperability to model transformation is broadly addressed in EU-funded projects such as INTEROP [Dou07], ATHENA [ATHEN03] and UEML [Uem03].

### 2.4.2. Approaches using MDA and Model Transformation

Based on the work in [MLZ05], [MZ05], [ZM05], [RM06], a new research approach has been defined to address limitations regarding extensibility [TZD07]. The research extends the concern/view of control flow by other concerns such as collaborations, data processing and fault handling. Second, the framework extends the transformation approach for integration of two specific kinds of process models, but provides interoperability with process models realized in other languages than those two specified.

The view-based modeling framework [TZD07] (VbMF) is based on the concept of architectural views. An architectural view is a representation of a system from the perspective of a related set of concerns [IEEE00]. Each particular concern is (semi-)formalized by a respective meta-model. A meta-model specifies entities and their relationships that appear in the correspondent view. VbMF defines a number of meta-models that conform to a common meta-meta-model (see Figure 16 (a)). This way, VbMF separates process concerns into a number of architectural views. Furthermore, VbMF exploits the model-driven architecture approach [OMG02], [VS06] to separate the platform-neutral models from the platform-specific models. For this reason, VbMF also separates process models into different levels of abstraction. A meta-model at a lower abstraction level is defined as an extension of the meta-models at higher levels. VbMF's meta-models are either directly or indirectly derived the Core meta-model (shown Figure 16 (b)) and therefore their relationships (aka trace links) are explicitly maintained via the model-driven architecture. These relationships enable VbMF to bridge the gaps between meta-models at different abstraction levels and to propagate changes. VbMF is able to generate code in executable language that can be deployed on existing process engines.

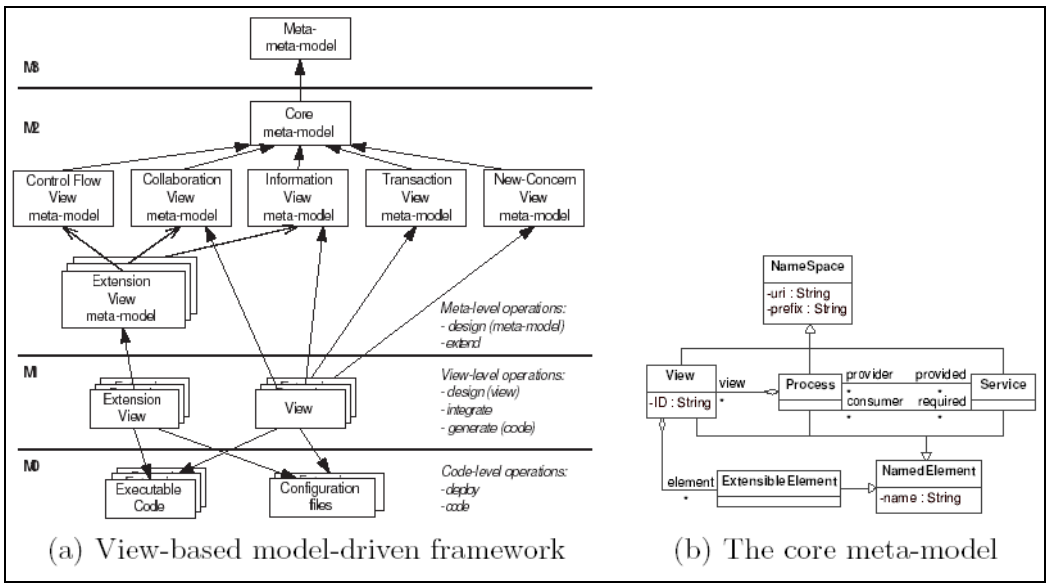



**Figure 16: The VbMF modeling framework and the Core meta-model**

The presented method [TZD08] also exploits the model-driven software development (MDSD) paradigm [VS06] to separate the platform-neutral views from the platform-specific views so that the business experts " in their views " can get rid of technical details. Platform-specific models or executable code, for instance, Java, or BPEL and WSDL descriptions, can be generated from the views by using model-to-code transformations. The separation of view abstraction levels helps in enhancing the adaptability of the process-driven SOA models to business changes. For instance, the business experts analyze and modify the abstract views to meet the requirement of changes. Then, these modifications can be transformed into code in executable languages. The technical experts work with platform- specific views to define necessary configurations such that the generated code can be deployed into the corresponding runtime (i.e., process engines and Web service frameworks).
In the context of process-driven modeling, there are a number of standard languages in which some provide high-level descriptions, for instance, BPMN [OMG06], EPC [Kin04], [vdA97b] and Abstract BPEL in WS-BPEL 2.0 [OASIS07]. EPC and BPMN provide high-level diagrams that consist of graphical notations for visualizing representations of processes. These diagrams are relevant to the business analysts.

SoaML by OMG [OMG09b] is proposing a modelling solution of SOA relevant technical questions such as web-service choreography or interfacing. This is based on UML2 and is integrating MDA principles. BMM can be used to formalize strategic aspects but the integration is not very detailed. It is obvious that OMG comes more from the technical modelling world. To close the gap between strategy and more technical modelling, OMG has planned to integrate BPMN as a process modelling notation into the SoaML meta-model.

In [TSD08] they argue that there is no explicit link between these languages and the executable languages. This has meanwhile evolved. Today, BPMN and EPC have formal meta-models and transformation is possible. Furthermore, a complete method should also integrate strategic modelling. It is true, that other approaches than [TSD08] might be a bit less efficient, but they still work. The presented case study in [TSD08] called "shopping process" is also not detailed how to derive automatically from Macro-Micro Flow technique in UML activity diagrams into BPEL and WSDL.

Two actual research teams are also dealing with model transformation. Stein for instance [Ste08] has shown through a case study an automatic transformation from EPC to BPEL based on workflow patterns. Stein has included validation steps with model checker technology with the objective to check if models satisfy a given temporal statement. The BPEL process was deployed on ORACLE SOA Suite, whereas Microsoft BizTalk failed. Issues on translation bugs for XML Schemas, integrating business rules into EPC without using XPath (technology dependent) were identified. As conclusion, the top down approach worked well, but a roundtrip scenario is almost impossible.

Another initiative presented in [ATHEN06] is about the PIM4SOA Meta model [BL06], which allows model transformations using the MDA principles. The meta model is arguing to enable the exchange of business process specifications between modelling tools and between tools and execution environment. MAESTRO, ARIS EPC and EXPRESS can be interfaced to PIM4SOA, which is then providing a link to web service integration (XSD, WSDL, BPEL).

Thomas [Tho07] has developed a process driven SOA based on EPC, BPMN, BPEL and WSDL. The reasoning is also top-down, based on a semi-formal approach. The EPC is used as an information model in a semi-formal graphical language. The configuration level is solved with the BPMN including process logic and technical details for the execution. The

execution level contains information for the execution that can be expressed e.g. by BPEL. WSDL is used to describe the web services. The research is illustrating a creative manual process of translating EPC-Model into a BPMN model. Reference models and patterns can be used for model construction. The semi-formal intermediate result through BPMN needs to be transformed to a BPEL model. BPEL is considered as the de-facto-standard for business process implementation based on web services. Transformation rules assign each BPMN element and attributes to BPEL representation. However, it is not fully automated as sometimes it is necessary to adjust the BPMN process design according to execution constraints required by BPEL compliance with the conceptual determining factors. Thomas et al. [Tho07] argue that a robust tool support is needed to allow graphic modelling (SemTalk for MS Visio or Intalio) and to enable BPMN to BPEL translation. Integration Platforms e.g. BizTalk Server with embedded tools like Visual Studio allow import, creation, processing and export of service orchestrations. Unfortunately, the research takes not into consideration how to derive from strategy the business requirement model EPC, but the approach illustrates for the process and IT level besides UML the most common languages used in a process oriented SOA implementation. The transformation mechanisms are not explained in detail, but it is obvious that EPC2BPMN is semi-automatic or manual, whereas BPMN2BPEL or BPMN2YAWL [DDDG08] is mostly automated process.

Finally, the latest approaches on languages and transformation from one level to another can be recapitalized in the following conceptual model indicating the sources names for the proposed solutions. An issue might still be the separation of process models concerns, explicit relationships between abstract and executable modelling languages. An additional manual effort to maintain the integrity, consistency and validation of models is necessary for semi-automated and manual approaches:

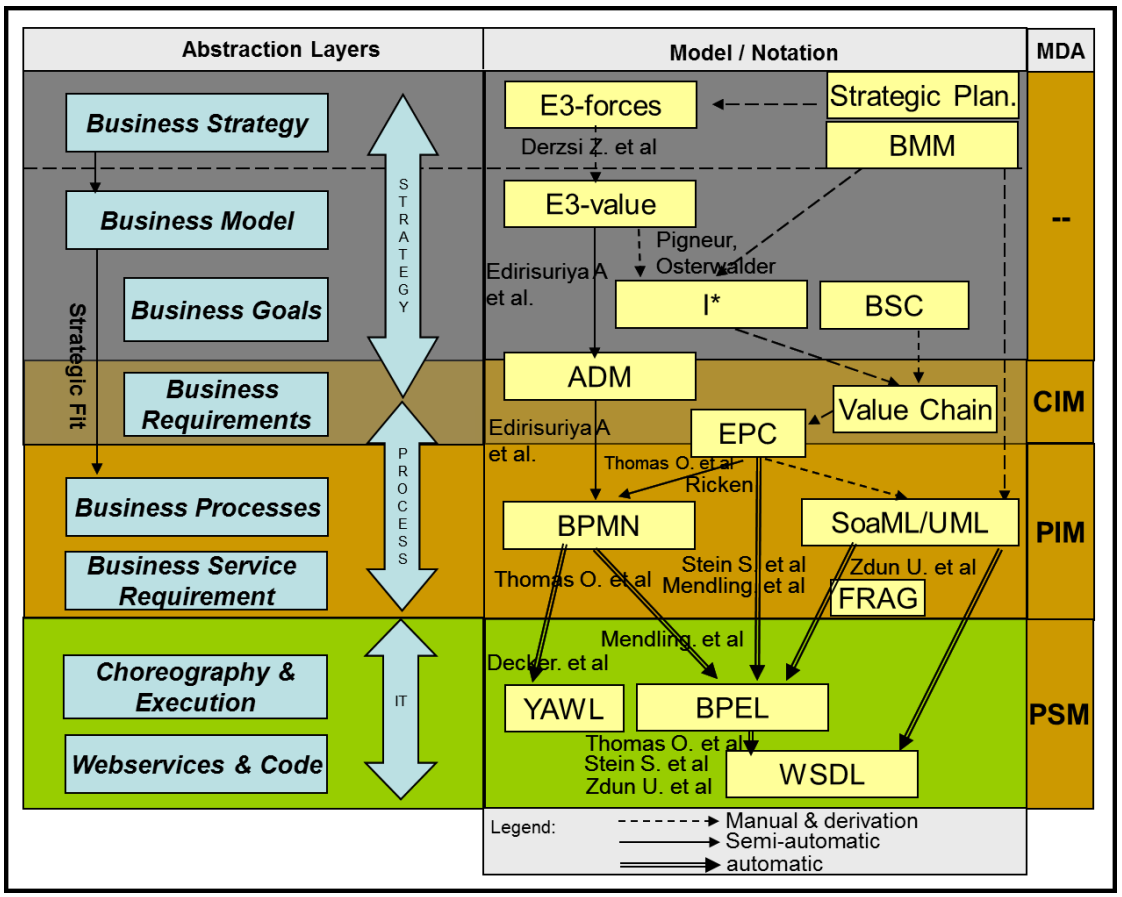



**Figure 17: Overview of Transformation Mechanisms between models/notations and Abstraction Layers**

### 2.4.3. Summary on Model Transformation

Model transformation and interoperability is a complex topic with a lot of actual research issues. Basically, there are different ways to approach the issue: First, in the context of top-down modelling for SOA, the question of meaningful transformation on what level needs to be resolved (Strategy2CIM, CIM2PIM, PIM2PSM). Due to the fact of a broad range of available process models, interoperability and reusability becomes an important question. Either direct transformations between specific process models are defined and used or a broader approach based on meta-models is used. To use the concept of different views and related concerns is obviously a good way to deal with different aspects other than just the control flow.

The strategy layer should not be neglected despite the fact that automation into process models seems not to be possible today. Within the strategic level, strategic planning turned out to be the most used approach to formalize strategy. The BSC deals with strategic objectives and related activities that can be directly linked to the value chain. To formalize business model, e3forces and e3value can be used. Both have a slightly different viewpoint as discussed in section 2.2.7. The I* stands on the same level as a BSC Model, and the link between e3value and I* is possible. It is very crucial to understand the different viewpoints and concerns for the models presented in the strategic layer.

As the concluding figure in this section illustrates, different path can be used. A valid question for practitioners is the ability of tools to cope with these transformation rules and to make sure the identified languages are consistent between each other. In terms of pure transformation objectives and transformation efficiency, MDA principles can be applied, which means also the usage of the UML notation family. It is imaginable to describe strategy with the BMM model and link to UML Activity diagrams or BPMN directly. But as this is not the only and first criteria, because UML has also shortcomings on strategy level and CIM level, the choice need to be designed related to the specific context the method will be applied in. This will be further elaborated in design rationales for modelling notation choices and used in the application cases.

### 2.4.4. Patterns for SOA construction support

The pattern movement is a software engineering success story. In 1995, the Gang of Four published their seminal Design Patterns book [GHJV94]. Many different types of patterns have been published since then, for example Patterns of Software Architecture (POSA) [BMR+96], domain analysis patterns, and even patterns for non-IT topics. Examples for recent contributions are Patterns of Enterprise Application Architecture (PoEAA) [Fow02], messaging [HW03], remoting [VKZ04], and SOA [HZ06],[ZHvdA06].

In this context Zimmermann et al. [ZZGL08] state that “a pattern is a proven solution to a problem in a context, resolving a set of forces”.

These researchers are explaining in their work [ZZGL08] that “the context refers to a recurring set of situations in which the pattern applies. The problem refers to a set of goals and constraints that typically occur in this context and influence the pattern’s solution, called the forces of the pattern. To systematically explain how to apply a number of patterns in combination, many pattern authors document patterns as part of larger pattern languages,

containing rich pattern relationships and extensive examples and known uses sections. Patterns in a pattern language are applied in an incremental refinement process. The decision making in this process is based on the pattern's forces". In the architectural realm, these forces include non-functional requirements and software quality attributes. Mostly, a compromise needs to be found to balance the forces. Zimmermann et al. [ZZGL08] argue further that "the pattern describes how the forces are balanced in the proposed solution, and why they have been balanced in the proposed way. In addition, the advantages and disadvantages of such a solution are described as consequences. Applying patterns during software design requires a broad view on how to select from a large body of patterns possibly eligible for a particular domain. Patterns do not focus on a single, domain-specific solution in a particular business context, but on generic design knowledge." For instance, the INVOKER pattern [VKZ04] describes how a middleware invokes remote objects in general. The pattern applies to all kinds of middleware, but does not explain the specifics of INVOKERS in a particular application context such as a specific SOA middleware implementation.

To illustrate the approach of using patterns, a case study worked out by [ZMCO04] was suggesting six steps [BMR+96] to implement the BROKER pattern. The case study reports about the issue to connect retail banks with a shared core banking backend provider. To resolve technology mismatches between the heterogeneous systems, a SOA concept based on web services has been implemented.

By definition, patterns are not the documentation of an individual system, but one source of (reusable) architectural knowledge to be considered and brought to bear when architecting a system. Therefore, applying a pattern is making a decision; the consequences of applying a pattern engender more decisions. This links patterns to a decision model called ArchPAd developed by IBM researchers [ZZGL08]. This is an architectural pattern-and decision-based design method. They propose 4 refinement stages:

"The first stage deals with requirements analysis and executive decisions as entry points into the architecture design work. The motivation for this stage is that some non-technical analysis and planning has to happen before any technical patterns can be applied. Executive decisions reside here. The runtime platform and programming language

In Stage 2, conceptual decisions are made; architectural patterns appear as AD alternatives. For instance, BROKER is an architectural pattern; deciding for or against it is a related conceptual decision.

In Stage 3, technological decisions are made and detailed design patterns are selected. For instance, the six implementation steps in the BROKER pattern as described in [2BMR+96] fall into this stage.

In Stage 4, implementation and deployment related decisions are made. Discrepancies between abstract concepts and implementation reality can be discussed and documented here – e.g., vendor products often implement a conceptual pattern in a specific way, have limitations, or offer proprietary extensions. Asset-level application server, workflow engine, and other middleware selection decisions fall into this stage, e.g., to use a particular SOAP engine for XML messaging in a Web services-based BROKER."

The case study was looking for the application of IBM's best practice by reusing architectural decisions gathered during the different phases in various projects. Additionally, a tool has been developed to integrate the gathered best practice architectural decisions [ZGK+07] This

approach developed by IBM researchers is for sure an added value for a top-down SOA method and will be considered as an enabler for the SOA domain model.

### 2.4.5. Introduction into top-down model-driven SOA Method

This section will show exemplarily one scenario for a process-oriented SOA implementation. The objective of this case consists in introducing the reader to the principles of process-and model orientation with a top-down design strategy as introduced in the first chapter as one of the major concerns to address.

The case illustrates a fictive company looking for improvements in the order-to-cash process which is a process that exists in all companies world-wide. The chief executive officer (CEO) of CaseStudy INC. has the wish to improve the order-to-cash process because of internal complaints from the financial division regarding the delay of sending invoices and the related credit collection. A project with external consultants found out, based on an industry benchmark, the cause for this delayed invoicing process was manual work and a non-standardized billing process. Furthermore, no exception handling was in place to deal with decisions on billing with a need to escalate to the top. The IT systems were also not able to support in the best way the process as a lot of manual work and corrections were necessary.

Due to the innovative product and high level of service quality, the company has grown very quickly without being able to update and reorganize the complete IT infrastructure and to make sure that the business processes can be supported in an efficient way. Another weakness identified, was the missing organizational link between production and finance. In order to address all these issues (concerns), a SOA has been proposed to the stakeholder. A portal should consequently be built to offer easy-to-use screens following the process logic through different business divisions. The financial system and the production planning system currently used can be re-used without being forced to purchase and integrate completely new systems. Available and needed functionalities were analyzed and covered the requirements for the new and improved process. The organization with a historical grown culture should step by step merge from the “silo mentality” to a more common view on processes with well-defined interfaces and responsibilities.

A program to build an Enterprise Architecture has been decided including Governance programs for processes, data, systems and the architecture. Following the project deliverables for the implementation phase, the cost savings calculations indicated a Return on Investment (ROI) after three years. The CEO decided in his company strategy for the next three years to come to increase the order-to-cash process dramatically, otherwise the financial division would not be prepared for the future growing rate of revenues (expected to be 11% p.a.).

For the strategic business model, several methods could be used. In this introduction, the Balanced Scorecard (BSC) [KN92] [KN93] model has been chosen.

The four perspectives of BSC can be represented in a so called “cause-and-effect” diagram. In the cause-and-effect diagram the necessary objectives for implementing a business strategy are defined and their mutual influence is depicted using a cause-and-effect chain running over perspectives.

The strategic objectives defined for the Case Study Inc. for the next 3 years following the method of BSC can be shown in figure 18 with a tool-driven approach in a diagram showing the relationships between the strategic objectives:

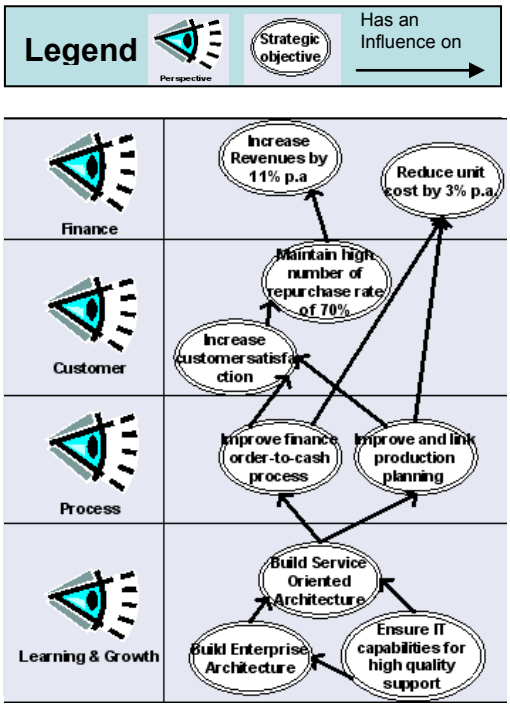



**Figure 18: Strategic Objectives in cause-and-effect chain**

In base level, within the Learning & Growth view, three objectives can be found, whereas "Ensure IT capabilities for high quality support" has an influence on the other two objectives. Because of SOA principles, processes can be improved ("Finance-to-Cash" and "Production Planning"). These two improved processes will impact the customer satisfaction positively, which means a high number or re-purchase rate. Finally, this has an impact of revenue increase. The improvement of processes has also an immediate effect on cost reduction objective. This type of (strategic) argumentation structured into a model is understandable for executives and therefore a good communication and formalization possibility.

Each strategic objective is measured with Key Performance Indicators (KPI) and is related to activities and/or projects to make the strategic objective happen. The connection down to the abstraction level of processes can be made by linking the strategic objective to the process (e.g. order-to-cash) in the process landscape.

The next deeper layer describes the design of business requirements in the form of a process model. This view provides a high-level insight into the general operations of a company. The high-level overview (figure 19) can be shown by a value-added chain diagram (VACD) and specifies the functions in a company which directly influence the real added value of the company [Por85]:

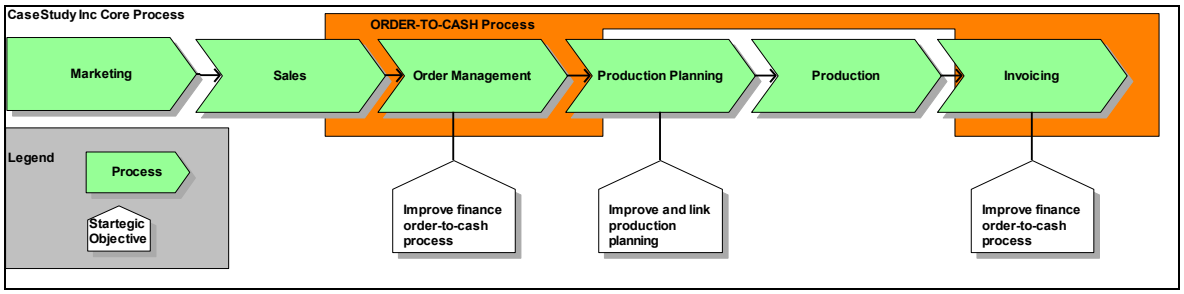



**Figure 19: Strategic Objectives linked to business functions**

In this specific case study the strategic objective "Improve order-to-cash process" is directly linked to the processes "Order Management" and "Invoicing" which is part of the primary activities or core processes.

EPC's have been promoted by Scheer [Sche93] and are used to represent the „procedural organization"[STA05] of the company, i.e. the links between the objects in the data, function and organizational view and, as a result, the processes are represented. The procedural sequence of functions is represented in process chains. In this context the start and end events of every function can be specified. Events trigger functions and are the results of functions. The conceptual foundation of EPC is based on Petri-Nets and PERT diagrams.

To allow more complex information on workflow, used data as input and output as well as who is carrying out the functions and with the help of which systems, the EPC model illustrates the details of the "order management" function. As described in the strategic objectives, we want to improve the process by replacing manual workload by automatic and intelligent web-services. The business rules are modelled as operators allowing in this example to formulate a decision path.

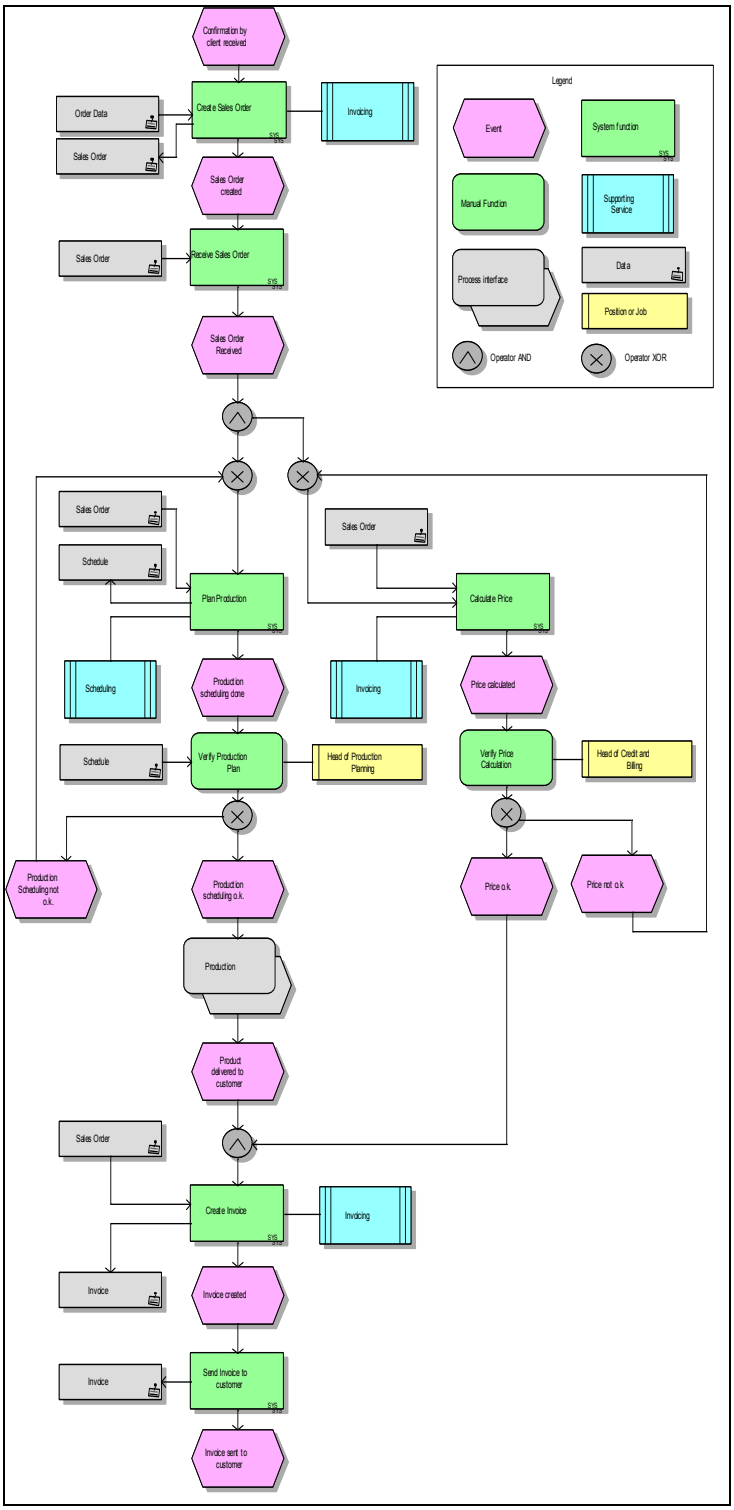



**Figure 20: EPC Model explaining the "To Be" Order Management process**

Figure 20 illustrates the simplified billing process from a service oriented angle: events formulate a decision in a certain point of time. Once the contractual details are fixed, the sales order is created and sent by sales to the department billing & collection department (finance) as well to production planning department. This is an incoming message for the finance department and triggers the price calculation service and the production planning service. Both functions are performed with the purchase order as entry and are supported by the scheduling service and the invoicing service. For each, a manual control is done by the divisions' supervisors. If production planning is fine, the production process is triggered.

Once the product is delivered to customer, the invoice is created and sent to the customer (outgoing message).

In the next step, the EPC process model is translated focussing on relevant information into a BPMN model. This is necessary to add more technical information to the model. The logical workflow is going through the pools of sales, finance (billing & credit collection), production planning and production. Within those pools, actors will perform the different tasks. The pools are separated by lanes.

The BPMN standard developed by BPMI [BPMI03] specifies a graphical notation that is foreseen serving as a common basis for a variety of business process modeling execution languages. The BPMN notation has been taken over by OMG and has since that time strongly evolved (further details in chapter 2.2.5.).

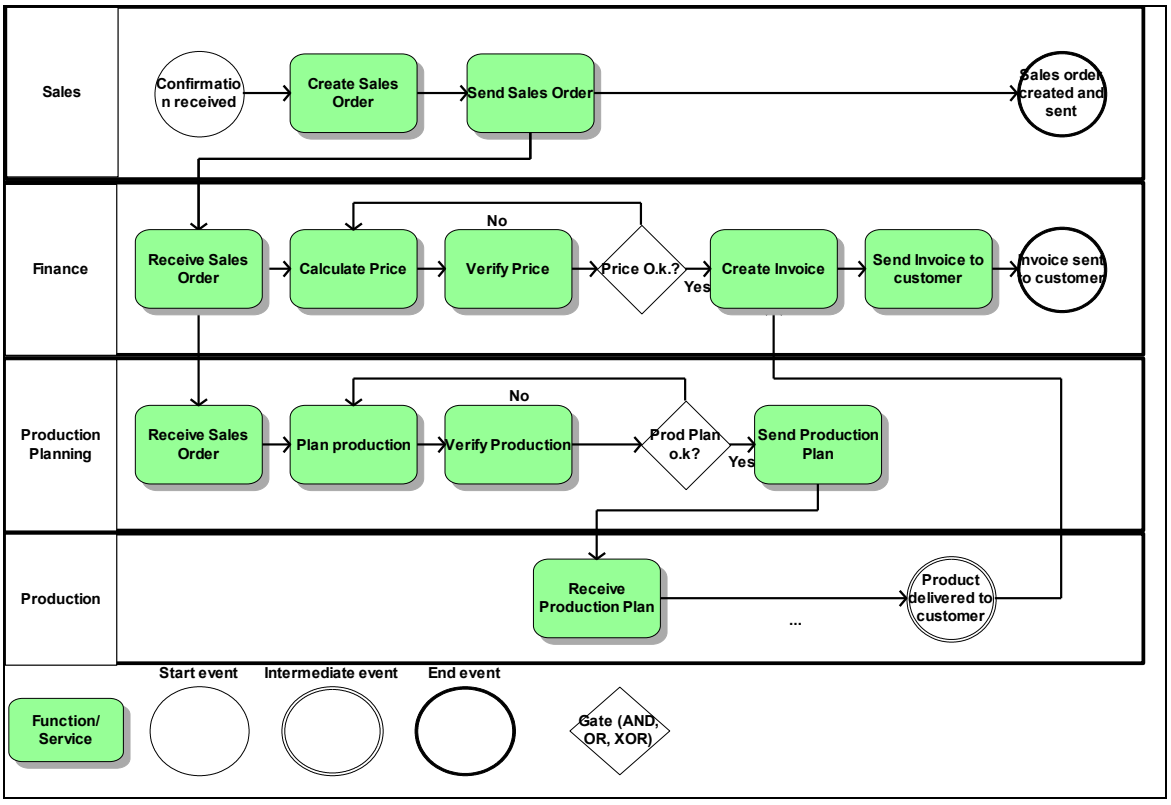



**Figure 21: BPMN Model explaining the "To Be" Order Management process.**

Important is the definition of incoming message and outgoing message between the pools. If the granularity of the service is too high, then decomposition is necessary. Different types of sub-processes can be identified: "embedded", "independent" and "reference". This needs to be done with the goal to end on a level of granularity that allows re-using or creation of new services.

The project team of CaseStudy Inc. decided to focus on the billing process as this was one of the strategic objectives. The BPMN model helps now to identify the services required for the execution of this process. One or more web-services (depending on granularity) can be assigned to an activity. As the process is executed horizontally throughout the company, different business divisions are involved which means that the improvement objectives are not

so obvious to achieve. Portal architects know after BPMN analysis, what services need to be called from the user portal.

Another goal, but not less important, is to ensure that XML languages designed for the execution of business processes, such as Business Process Execution Language for Web Services (BPEL4WS or BPEL), can be visualized with a business-oriented notation. As the utilization of re-usable services is a key criteria in SOA, the Web Services Description Language (WSDL) will be used [W3C01], [Chi04]. WSDL is an XML-based, platform independent meta language used to describe the interface definitions of a Web service. In WSDL, the externally accessible functions of the Web service and the parameters and return values of these operations are defined. WSDL describes the communication format in which function calls to Web services are transmitted.

BPEL links WSDL descriptions into a logic process flow. A BPEL process is according to this logic a bunch of service executions in a logical and timed sequential order. This is also well known under the term "service orchestration" [Bla03].

Following the project plan, BPEL Code with the integrated WSDL service description needs to be implemented to make a process-driven SOA happen.

However, the reality in CaseStudy Inc. is that fully automated processes represent only a small fraction of the processes that are actually executed. Most comprise manual activities that must be carried out by staff e.g. the manual controls of price and production plan. In a later phase, this could eventually be automated. A further problem is the performance of complicated calculations or data transformations that are necessary for preparing or processing the data used by the invoked web service. This issue needs to be solved by the IT developers in the project.

The strict and deterministic Rules in BPMN allow the automatic generation of BPEL Code. Tools with the ability to transform BPMN models into BPEL code (e.g. Visual Studio with its integration platform BizTalk Server from Microsoft) are able to import, create, modify and orchestrate web-services.

The case study focused just on specific aspects for a model-driven SOA implementation. In such projects, many other decisions in areas like adapters, security, orchestration, technical communication, transformation etc. need to be addressed.

Figure 22 is summarizing the decisions on standards used on the different level of abstractions:

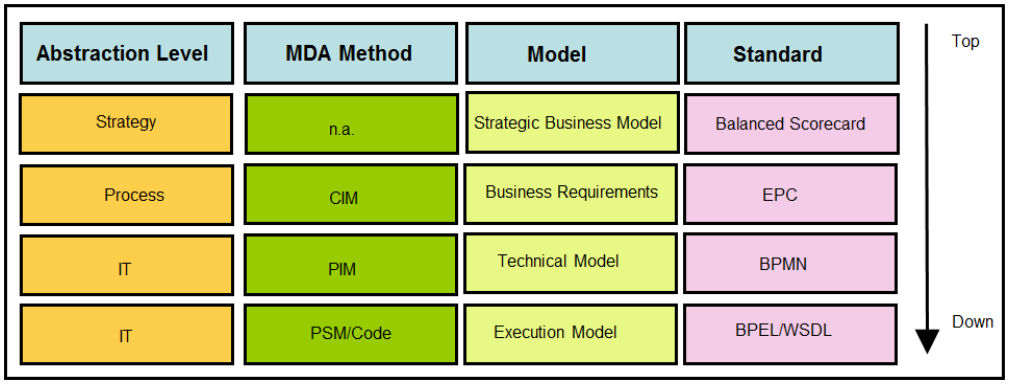



**Figure 22: Models & Standards for the introduction Case Study**

The dimension of data modelling has not been focussed on, as in general data modelling is more advanced and mature than strategy or process modelling notations. In general, data management is defined as [BD07]:

**Definition 32: Data Management [BD07]**

> *"Data Management is the framework of processes and technologies aimed at creating and maintaining an authoritative, reliable, sustainable, accurate, and secure data environment that represents a "single version of truth," an accepted system of record used both intra-and inter-enterprise across a diverse set of application systems, lines of business, and user communities."* [BD07]

The fictive, but practical example has illustrated a top-down approach for SOA implementation without using a particular SOA Method. A particular concern that needs to be addressed is the model transformation mechanisms between the modelling notations on different levels of abstractions.

## 2.5. SOA Methods and Frameworks

### 2.5.1. Introduction to SOA Methods

Traditional software engineering methods are simply not adapted any more to the changed requirements related to modern SOA implementations. Especially a risk of failure is the increased number of stakeholders with potential conflicting business needs [Ars04], more and more dynamic environments and issues of decision-making between design-time environments and run-time environments need to be considered.

Novel techniques must be developed to support the refinement from the early phases of requirement analysis to the final steps of implementation and deployment. Similarly, novel techniques must be devised to construct compositions of web services that at run-time can provide feedback and significant information to business analysis and stakeholders, who can use this information to devise new business strategies or take strategic decisions at design time [IEEE00].

The lack of method for SOA construction, identified by Papazoglou, Traverso, Dustdar and Leymann in the "Service-Oriented Computing Roadmap" [PTDL06] as the main challenge for SOA, is a key academic driver for the present thesis.

Therefore, this thesis will propose a coherent method to implement SOA using a situational ME approach, taking into account strategic business aspects. The expectations on such a method will depend of the enterprise context, enterprise size, enterprise industry, etc. The enterprise context (e.g. the financial situation, enterprise culture, IT maturity and IT competencies) will drive the expectations towards a SOA implementation. The enterprise size will also have a big impact on the potential savings that can be achieved, whereas the enterprise's industry and the business model will also influence the expectation. Generally speaking, the bigger and the more complex a company and the supporting IT is, the higher the

probability to get the full benefit out of SOA because candidate services can easily be re-used in different places.

The objectives of this first steps are to (i) oversee SOA implementation methods with their capabilities (ii) to identify a list of sub-domains to consider for a SOA implementation (iii) to summarize the identified SOA sub-domains into a SOA Domain Model.

Before starting to explain SOA implementation methods, it would be necessary to briefly explain what a method is [Cre98]:

> "the analysis of the principles of methods, rules, and postulates employed by a discipline". [Cre98]
>
> or
>
> "the development of methods, to be applied within a discipline"
> "a particular procedure or set of procedures". [Cre98]

Creswell [Cre98] is stating that "Method refers to more than a simple set of methods; rather it refers to the rationale and the philosophical assumptions that underlie a particular study." and "Another key (though arguably imprecise) usage for method does not refer to research or to the specific analysis techniques. This often refers to anything and everything that can be encapsulated for a discipline or a series of processes, activities and tasks."

Methods in and of themselves are meaningless without clear expectations. Expectations can include terminology definition, process or procedure guidelines, etc. It will not matter how a problem is approached, if the expectation on the method was not defined and its achievement evaluated, the solution is worthless.

Therefore, it is proposed to introduce another definition focusing to the way of solving a problem:

**Definition 33: Method [Ver92] [Bri96]**

> *„A method is a set of methods, models and tools to be used in a structured way to solve a problem."* [Ver92]
>
> and
>
> *"A method is an approach to perform a systems development project, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding development products."* [Bri96]

With these definitions in mind, we will analyze different proposals and evaluate them in a structured way.

## 2.5.2. Methods for SOA Implementation

To be able to propose a sound method, it is first necessary to analyse and structure available SOA implementation methods. Therefore, a set of approaches known in academia and practice have been carefully analysed.

Generally, the listed methods have very different viewpoints and focus such as high-level vs. detailed, functional vs. technical, academic vs. best practice, non-profit vs. profit organization and top-down vs. bottom-up. These viewpoint(s) depend strongly of the origin of the author.

The table 12 gives an overview of available approaches identified as relevant for SOA and studied in general in alphabetical order:

**Table 12: Methods for SOA implementation**

| Name of SOA Method | Shortname | Author(s) | Organization/Date | Model Notation Mentioned |
|---|---|---|---|---|
| Architecting Industry Standards for Service Orientation [Lee05] | - | J. Lee | Microsoft, 2005 | No |
| ARIS Value Engineering for SOA [Yva06] | AVE for SOA | Konstantin Yvanov | IDS Scheer AG, 2006 | EPC, BPEL, WSDL |
| CBDI-SEA SOA Reference Framework [But09] | CBDI-SEA | John Butler | CBDI Journal, 2007 | BPMN, UML Activity |
| Enterprise SOA [WM06] and update with Accelerated Transformation to SOA [SAP08] | E-SOA | Dan Woods, Thomas Mattern | SAP AG, 2006 | BPMN |
| Enterprise SOA Adoption Strategies [Jon05] | - | Steve Jones | Capgemini 2006 | No |
| Model-Driven Integration of Process driven SOA Models [ZD06] and View-based Integration of Process-Driven SOA Models AT Various Abstraction Levels [TZD08] | Based on MDSD | Zdun & Dustdar<br>Tran, Zdun, Dustdar | Distributed Systems Group, 2006-2008 | UML Activity Diagram, DSL, FRAG |
| Platform-independent model for service-oriented architecture [BL06] | PIM4SOA | Xabier Larrucea et al, ATHENA Project | European Software Institute (ESI) Spain, DFKI GmbH Germany, SINTEF ICT, Norway, 2006 | UML |
| Service-oriented Design and Development Method [PvdH06] and [NvdHP+09]. | SoDD | Mike P-Papazoglou | University of Tilburg. June 2006 | BPMN, BPEL, WSDL |
| Service oriented Modeling & Architecture [Ars04] and [AGAAG+08] | SOMA | A. Arsanjani | IBM,2004 | UML Activity Diagram, BPMN, BPEL, WSDL |
| ORACLE Unified Method for SOA [ORAC11] | - | - | ORACLE, [ORAC11] (former BEA: SOA practitioners, [Shu06]) and [SUN04] | VACD, UML; BPMN, BPEL, WSDL |
| SOA Reference Model for Service Oriented Architecture [Mac06] | - | MacKenzie et al | OASIS,2006 | No |
| SOA Adoption Model [GART06] | - | - | Gartner, 2006 | No |
| SOA Delivering Strategies [Erl05] | - | Thomas Erl | SOA – Concepts, Technology and Design, Cahpman Hall 2006 | UML, BPMN, BPEL, WSDL |
| SOA Organizational Roadmap [KBS06] | - | Dirk Krafzig et al | Enterprise SOA book, Coad Series 2006 | UML, BPMN, BPEL, WSDL |

The detailed analysis of selected SOA methods is presented in section 3.2. once the SOA domains and sub-domains have been defined and explained. Furthermore, Annex B is reporting on the method details.

## *2.6. Method Engineering for SOA Method construction*

### 2.6.1. Introduction to Method Engineering

Flexibility without control can hardly be considered a method, since any systematic and coordinated approach to establishing work methods is absent. For such an approach to be systematic and coordinated requires the technique of ME [BLW96]. This section is introducing the details about ME and is preparing the ground for chapter 4 for the creation of research artefacts. First, situational ME is explained. Second, the formalisation mechanisms for process fragments are explained.

### 2.6.2. Situational Specific Method Engineering

ME and Situational Method Engineering (SME) in general propose a way to formalize how method can be used for various developments. Following to Brinkkemper [Bri96], ME "is defined as the engineering discipline to design, construct and adapt methods, techniques and tools for systems development." Following to Henderson-Sellers and Ralyté [HSR10], SME can be considered as a component of ME, which "encompasses all aspects of creating a development method for a specific situation." During the analysis of available methods for SOA implementation, methods turned out to be not complete, difficult to apply and simply not flexible enough. Therefore, SME seems to be the solution to the problem of an engineering method being the best appropriate for specific organization with its SOA implementation project. The modular method construction as presented by ME allows selecting predefined method fragments that are matching to the context and objective of the project. All method fragments are stored in a method database. These method fragments are assembled using rules depending on the specific project decisions the project manager will go for. With this approach, an effective, efficient, complete, and consistent method for SOA implementation is intended to be achieved.

Engineering of a situational method requires standardized building blocks and guidelines, so called meta-methods, to assemble these building blocks [BRH99]. The importance of such methods was already recognized in 1991 by Olle et al [Oll91]. Their concept of "scenario philosophy" has been further elaborated by Kumar and Welke [KW92], who introduced method engineering, being an approach to develop and implement methods. Further research has been done to compare methods, the creation of a "method base" [HO93] or the introduction of task packages [Sae93] being part of the process perspective. Recently, some PhD work applying ME in various context have been realized e.g. for software product management [Wee09] [HR08] and requirements and architecture by Lehtola and Kauppinen [LK06].

A state-of-the-art review [HSR10] on SME is summarizing the "most significant challenges

- the rate of industry adoption and
- how to automate method construction process".

**Definition 34: Method Fragment [Har97] and [Bri96]**

> *"A method fragment is a description of an IS engineering method, or any coherent part thereof [Har97]." and*
>
> *" Method Fragments are distinguished in process fragments, for modelling the development process, and product fragments, for modelling the structure of the products of the development process [Bri96]".*

Once defined, this fragment is stored in a method database. When these method fragments are used for a specific project application, they need to match to the specific project characteristics. Rolland & Pracash [RP96] are arguing that further information is needed "for the usage context" of these fragments. Context is defined as a pair of "situation and decision" Therefore, the further description or knowledge of these method fragments is describing the situation and the decision in which the fragment could be relevant.

Mayer et al. [MCFKP+95] illustrate the ME process (using IDEF3 notation) as follows:

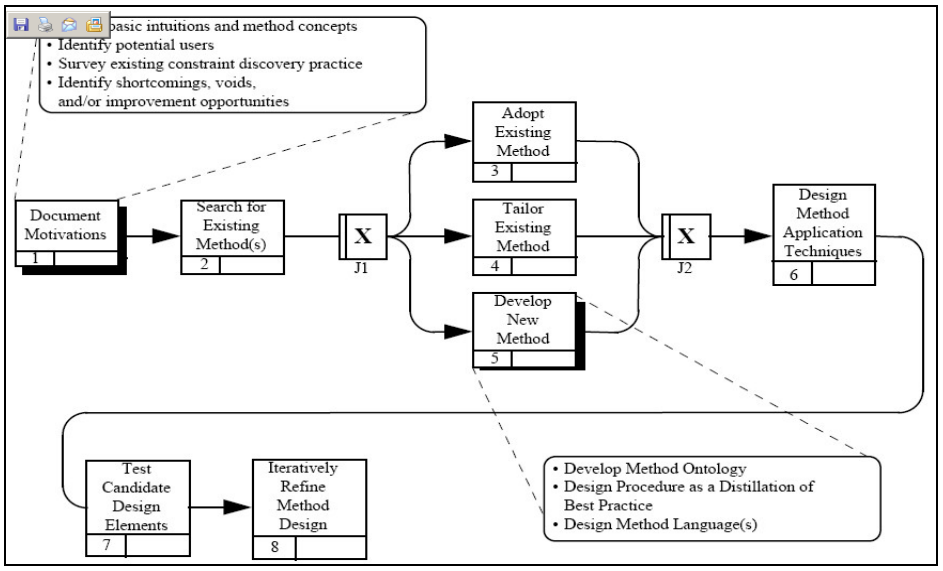



**Figure 23: Example of a Method Engineering Process**

Initially a motivation document should argue (1) on why a new method is required (specific user needs, shortcomings, improvement opportunities etc.) then (2) searching what methods are available to (3) re-use existing methods and/or (4) tailor them in order to satisfy needs or if adopting (3) and tailoring (4) cannot satisfy needs, then the development of new method (5) becomes necessary. The methods should then be designed (6) and (7) tested to finally iteratively refine (8) the method.

In the context of the present thesis, we will be in favor for the situational method as we need to apply the method to the project environment, context and objectives. Therefore, a configuration process is guiding the assembly of these building blocks into a situational method. Figure 24 illustrates this process and has been adapted from Brinkkemper [Bri96]:

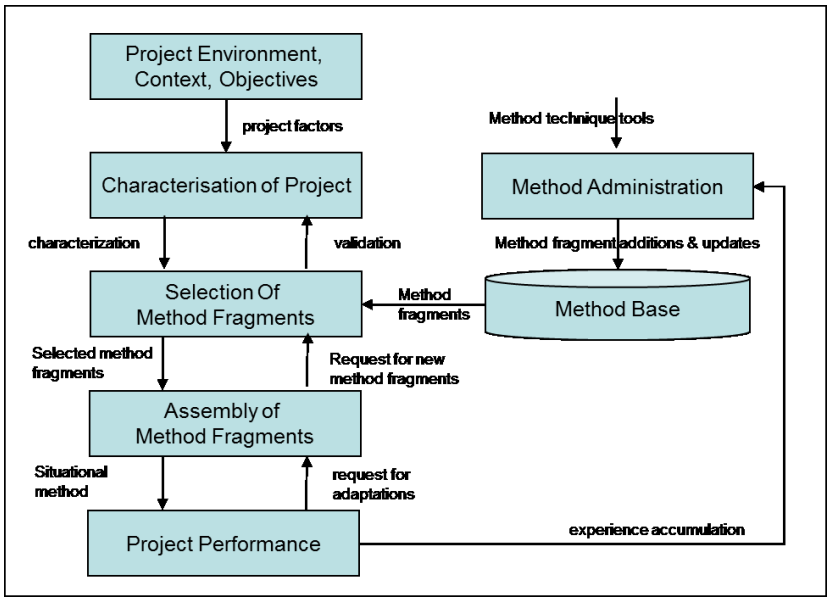



**Figure 24: Configuration Process for Situational Methods**

Method fragments can be distinguished into two kinds: product and process fragments. Product fragments model the structure of the products (deliverables, diagrams, tables, models etc.) of a SOA implementation method. Process fragments are tasks to support the solution path for issues or questions to resolve. In other words, process fragments represent phases, activities and tasks to be carried out. The term "method chunk" used by Harmsen [Har97] and Ralyté [Ral99] needs to be understood as the combination of a process fragment and a product fragment which are tightly coupled. In this present thesis we will stay with the term of method fragment as described in figure 25:

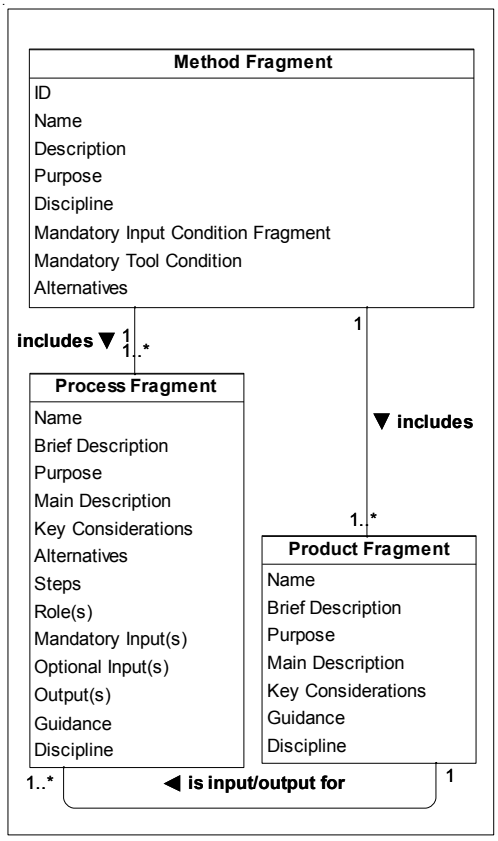



**Figure 25: Method Fragment is containing Process Fragment and Product Fragment**

Conditions or business rules are important for every process fragment, because they are specifying constraints [Bri96] and can influence strongly the process fragment or even stop it in some cases. A business rule is therefore a necessary or required condition or prerequisite.

The attributes will be explained in detail in chapter 4 when the alignment between the SOA Domain Model and ME is done.

ME and SME is a very broad field and actual research directions [HSR10] such as "how to best gather requirements and how to move from requirements to semi-automated or automated way of identifying the optimal collection of these fragments" are named. Furthermore, most of researchers have looked at creating method fragments from scratch, but how to formalize existing methods and to drive software vendors and consulting organizations to formalize and provide their methods in SME method fragments to cope with quality concerns such as consistency and completeness are so far not explored in detail.

### 2.6.3. Fragment specification and formalization

In order to create efficient representation of method fragments, the UML profile extending UML for the specific domain of fragment description is used. Software Process Engineering Metamodel (SPEM) [OMG08] is a meta-model and an UML profile that has been defined for standardizing software engineering process.

SPEM 2.0. is a highly formalized UML Meta-Model and is the "de-facto method" for ME application and therefore the meaningful parts out of this concept is used. The concepts of SPEM 2.0. related to "Method Content" and "Process" will be used to demonstrate the application. The SOA Domain model can therefore be considered as "input" information for the formalized application of ME.

SPEM 2.0. Meta-Model is compliant to MOF Meta-Model and is re-using UML2 notation. Furthermore, SPEM 2.0 is containing a Meta-Model and an UML profile as well. SPEM 2.0 separates reusable core method content from its application in processes. A development process defines the structured work definitions that need to be performed to develop a system, e.g., by performing a project that follows the process. Similar to SPEM 2.0., other ME meta models exist e.g. ISO/IEC 24744/2007 initially developed by Gonzalez-Perez and Henderson-Sellers [GPHS08] and could also be used instead. Using SPEM2.0. in this thesis is related to tooling on-hand with Eclipse Process Framework (EPF) [Eclipse09]. The tooling capabilities are further explained in section 5.3.

The following figure 26 contains the following classes, which are then further detailed in tables 13 to 17:

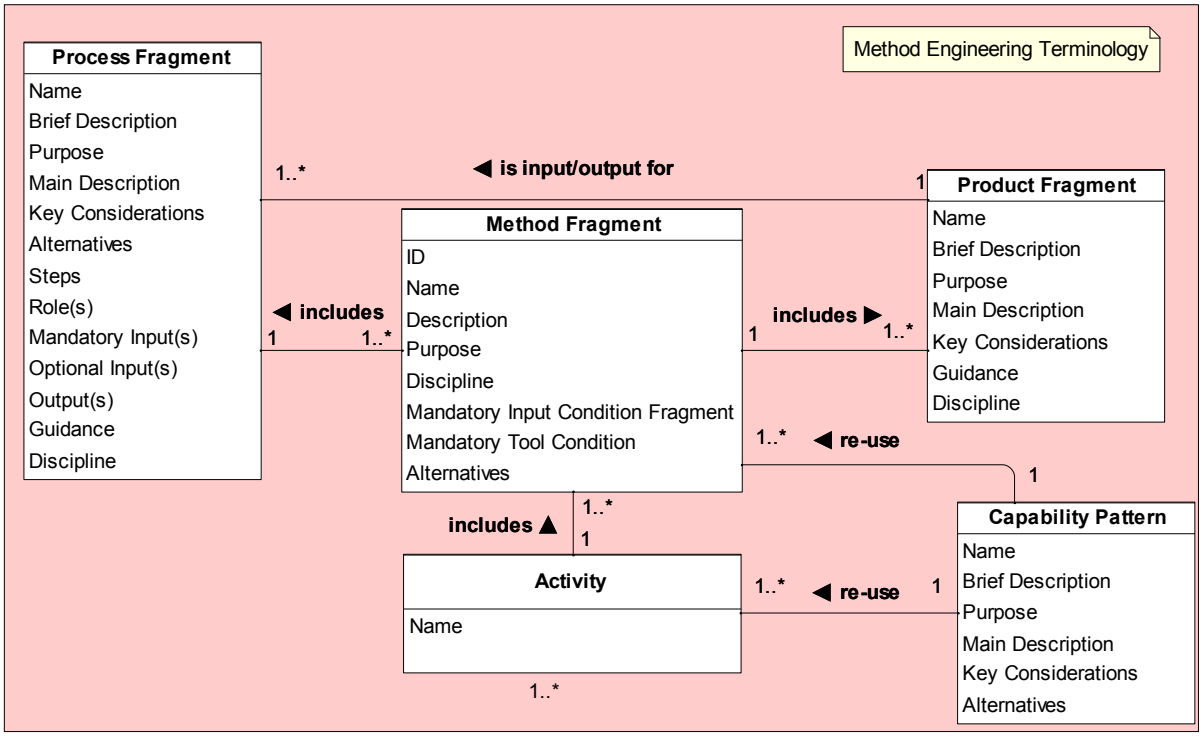



**Figure 26: Method Engineering Classes used**

Following to Eclipse [Eclipse09], a capability pattern is "a special process that describes a reusable cluster of activities in a general process area that provides a consistent development approach to common problems. Capability patterns can be used as building blocks to assemble delivery processes or larger capability patterns."

**Table 13: Attributes of Capability Pattern Class**

| Class | Attribute | Description |
|---|---|---|
| Capability Pattern | Name | Name of Capability Pattern |
| Capability Pattern | Brief Description | Short Description of Capability Pattern |
| Capability Pattern | Purpose | The purpose or why the Capability Pattern is used/ proposed |
| Capability Pattern | Main Description | Detailed description of Capability Pattern |
| Capability Pattern | Key Considerations | Further information of conditions, difficulties and context occurring |
| Capability Pattern | Alternatives | Alternate Capability Pattern covering similar requirements |

The capability patterns are a group of activities, which can be re-used in a specific context. The attributes used specify purpose, description, key considerations and alternatives that could be used. The work is re-using two patterns which are strategy modeling and CIM

modeling top-down. Therefore, these patterns can help ensuring easier re-use of best practice already identified as successful and efficient on other cases.

**Table 14: Attributes of Activity Class**

| Class | Attribute | Description |
|---|---|---|
| Activity | Name | Name of Activity |

The activity term is used to "neutralize" the specific semantic used for the fragments. It is necessary to provide a common or objective understanding of general activities. Activities are linked to one or more available fragments proposed in order to solution the underlying question raised. Following to Weerd and Brinkkemper [WB08], activities are used for capturing the process-view of a method.

**Table 15: Attributes for Method Fragment Class**

| Class | Attribute | Description |
|---|---|---|
| Method Fragment | ID | Acronym for fragment |
| Method Fragment | Name | Name is containing the ID followed by sequential number |
| Method Fragment | Description | Description of Method Fragment |
| Method Fragment | Purpose | Purpose of Method Fragment |
| Method Fragment | Discipline | Level of abstraction on which the fragment can be used |
| Method Fragment | Mandatory Input Condition Fragment | Fragment which is necessary/mandatory as input for selected fragment |
| Method Fragment | Mandatory Tool Condition | Tool required for selected fragment |
| Method Fragment | Alternatives | Alternate Method Fragments to consider. |

The method fragment is linked to activity and needs to be explained. A method fragment is containing process fragment and product fragment allowing the method engineer to understand, where this method fragment is coming from and to what discipline it is linked to. Mandatory input conditions, tool conditions are important to understand potential impact.

**Table 16: Attributes of Process Fragment Class**

| Class | Attribute | Description |
|---|---|---|
| Process Fragment | Name | Name of Process Fragment |
| Process Fragment | Brief Description | Short Description of Process Fragment |
| Process Fragment | Purpose | The purpose or why the fragment is used/proposed |
| Process Fragment | Main Description | Detailed description of process fragment |
| Process Fragment | Key Considerations | Further information of conditions, difficulties and context occurring |
| Process Fragment | Alternatives | Alternate process fragment |

| | | |
|---|---|---|
| | | covering similar requirements |
| Process Fragment | Steps | Detailed steps of fragment corresponding to work-step level |
| Process Fragment | Role(s) | Role(s) performing the process fragment |
| Process Fragment | Mandatory Input(s) | Mandatory Product Fragment input |
| Process Fragment | Optional Input(s) | Optional Product Fragment input |
| Process Fragment | Output(s) | Product Fragment Output(s) |
| Process Fragment | Guidance | Additional information which can be guidelines, examples, checklists etc. |
| Process Fragment | Discipline | Is a customized category, which can be tailored related to method engineer needs. |

The process fragment is part of the method fragment and describes the details of the process fragment with name, brief description, purpose, main description, key considerations, alternatives, steps, roles, mandatory input, optional input, output, guidance and discipline.

**Table 17: Attribute of Product Fragment Class**

| Class | Attribute | Description |
|---|---|---|
| Product Fragment | Name | Name of Product Fragment |
| Product Fragment | Brief Description | Short Description of Product Fragment |
| Product Fragment | Purpose | The purpose or why the fragment is used/proposed |
| Product Fragment | Main Description | Detailed description of Product Fragment |
| Product Fragment | Key Considerations | Further information of conditions, difficulties and context occurring |
| Product Fragment | Guidance | Additional information which can be guidelines, examples, checklists etc. |
| Product Fragment | Discipline | Is a customized category, which can be tailored related to method engineer needs. |

Similar to the process fragment, the product fragment is creating an artefact which will help to resolve the raised question. The product fragment is part of the method fragment and is input or output for the process fragment.

The following table is matching the terms between the ME definitions and definitions used in SPEM 2.0. This table is necessary to unambiguously identify corresponding definitions and to align on used semantics.

**Table 18: Terminology alignment table between ME and SPEM2.0.**

| **Method Engineering** [Bri96] [WB08] | **SPEM 2.0. Semantic/ Definition** [OMG08]: | **Example** |
|---|---|---|
| Actor | Role Definition | Business Analyst Role |
| Process Fragment | Task Definition | Model Service Oriented Business Process |
| Product Fragment | Work Product Definition | EPC Model |
| Role Instance | Role Use | Business Analyst xy |
| Process Fragment Instance | Task Use | Model Service Oriented Business Process in Project xy |
| Product Fragment Instance | Work Product Use | EPC Model xy |
| Work step | Step | Create and name EPC model |
| Rationale | Description | EPC is one of the standard Process Modelling notations on CIM-level and used for... |
| Activity | Activity | Model Business Requirements with EPC |
| Route Map | Capability Pattern | CIM Top-Down Modelling |

The example description is further explaining how these terminologies need to be understood.

## 2.6.4. Tool usage for situational SOA Method

A key step in the proposed configuration process is the selection of method fragments. These fragments need to be assembled and combined in such a way that the outcoming situational method does not contain any defects or inconsistencies. Several types of defects can appear [Hid93]:

- Internal incompleteness, which is the case if a method fragment requires another method fragment that is not present in the situational method. For instance, a data model has been selected without the corresponding modelling procedure and tool.

- Inconsistency, which is the case if the selection of a method fragment, contradicts the selection of another method fragment. For instance, two similar data modelling techniques have been selected without any additional reason.

- Inapplicability, which is the case if method fragments cannot be applied by project members, due to insufficient capability.

All these issues relate to the internal or situation-independent quality [HH95] of a situational method, i.e. the quality of a method without taking into consideration the situation in which the method is applied. The two most important criteria are [BSH99]:

- Completeness: “SME includes the method fragments that are referred or linked by other fragments in the situational method. Completeness is decomposed into: Input-Output completeness, content completeness, process completeness, association completeness, support completeness.”

- Consistency: “all activities, products, tools and people plus their –mutual-relationships in a situational method do not contain any contradiction and are thus mutually consistent. Consistency is decomposed into: Precedence consistency, perspective consistency, support consistency, granularity consistency, and concurrence consistency.”

In order to cope with the ME quality requirements for method assembly, it is necessary to use a Computer Aided Method Engineering (CAME) tool including a Method Engineering Language (MEL). This specific language should be able to support the method assembly (project characterization, fragments selection, choice validation, fragments assembly, situational method consistency check) and provide some generation engine (help & training facilities generator, tool infrastructure generator, project mgt. facilities, activity generator and project repository generator) [Bri96].

To current state, the tool MetaEdit+ developed by Kelly, Lyytinen and Rossi is not able to provide method assembly functionality yet [KLR96]. Mr. Juha-Pekka Tolvanen being the CEO of Metacase company confirmed on request that assembly techniques are not available in acceptable maturity to present state. Taking the non-availability of assembly tools neither the objective to create a domain specific language (DSL) in the present application case, a freely available and broadly used tool integrating SPEM2.0. was available at hand to be used as technical infrastructure for the application of method fragments. This will be explained in detail in chapter 6 tooling and prototyping.

### 2.6.5. Method Rationale in Method Engineering

According to Tolvanen [Tol95], “metamodels alone are inadequate to manage method refinements, because they cannot explain the evolution of a method”. Therefore, it is required to use method rationales. Method rationales occur at two different levels depending on the users. For method engineers, method rationale is an explanation why certain types or constraints of the method are included in the constructed method (method construction rationale).

As method users can understand method rationale differently, it is required to explain them a bit further. It can help to explain better, why a rational has been used. The details allow the method user more explicit and detailed understanding of the application and therefore the user can then better decide, if the rationale should be applied in his own context or not. Tolvanen is explaining that “the rationale of method use is normally not well documented because tools do not allow the capture of decisions about method use; only decisions about design choices. Therefore, it is the task of method engineers to collect the rationale of method use.” [Tol95].

Jarczyk et al [JLS92] explain that “a design rationale is the explicit listing of decisions made during a design process, and the reasons why those decisions were made.” Following to Horner and Atwood, [HA06] “the primary goal is to support designers by providing a means to record and communicate the argumentation and reasoning behind the design process.” Therefore, Lee [Lee97] explains what should be included in design rationales:

- the reasons behind a design decision and the justification for it,
- the other alternatives considered,
- the tradeoffs evaluated, and
- the argumentation that led to the decision.

The way how design rationales are represented, as the capture and usage should be as efficient as possible. Lee is arguing that three categories (formal, semi-formal, informal) exist. All type of recording are possible, but e.g. for formal design rationale, a computer must be able to read and process the design rational. If this is the case, the design rationale can hardly be understood by human beings [KR70].

In the present work, we will use a more semiformal representation which should combine the strengths of a formal and an informal representation. Semiformal representations try to combine the advantages of informal and formal representations. Lee [Lee97] is arguing that "on one hand, the information captured should be able to be processed by computers so that more computer based support can be provided. On the other hand, the procedure and method used to capture information of design rationale should not be very intrusive. In the system with a semiformal representation, the information expected is suggested and the users can capture rationale by following the instructions to either fill out the attributes according to some templates or just type into natural language descriptions." [Lee97]. Again, we stick to SPEM standard rationale descriptions, which are implemented as specific attributes in a formalization tool.

We will re-use ME later in section 4.2. for the purpose of formalizing available SOA methods into method fragments to populate the method fragment database.

## 2.7. Summary on state-of-the-art

The chapter on literature review was done with the purpose to elaborate and prepare the presented research questions posed in the first chapter. First, the viewpoint of a TD-MD-PO was explained in sections 2.1. to 2.4., second the objective of SOA Engineering Method was prepared through sections 2.5. to 2.6.

The chapter has introduced EA as starting point and underlined the viewpoint approach on different levels of abstraction. It was shown what EA methods are available and how the term "domain" has been defined and applied to the research subject. Modelling languages supporting the process-oriented approach have been analysed and classified on the different levels of abstractions. Interfaces between abstraction layers and model transformation mechanisms have been analysed and explained.

SOA and the term "SOA heartbeat" has been explained and defined in the introduction. Next, available SOA implementation methods were listed and will be qualified through the SOA Domain Model in chapter 3. Finally, ME with its basic concepts has been introduced to explain the value for an efficient SOA implementation method application for situational use as present or available methods do not offer a situation specific approach.

# CHAPTER 3

# RESEARCH CONTRIBUTION:

# SOA DOMAIN MODEL & SOA METHODS QUALIFICATION



Chapter three is detailing and focusing on the original research contribution of this thesis. First, an overview is provided by showing how the contribution chapter fits to the indicated research questions in the introduction and which artefacts are created. The value of these artefacts is described (section 3.1.). Then the details are explained in depth by starting with the SOA Domain Model (section 3.2.). The 5 domains of the main contribution "SOA Domain Model" are identified, defined and the SOA context is discussed. Once the domains are defined, they are used to create the SOA Domain Model which will then structure the analysis of selected methods. The methods are qualified and positioned through the SOA Domain Model.

Next, an evaluation of research artifacts by data collection through survey design is done to complete state-of-the-art with practitioners'' feedback (section 3.3.).

The following overview model in figure 27 is illustrating the big picture on the research contribution in chapter 3 (artefact 1 & 2) and chapter 4 (artefact 3 & 4):

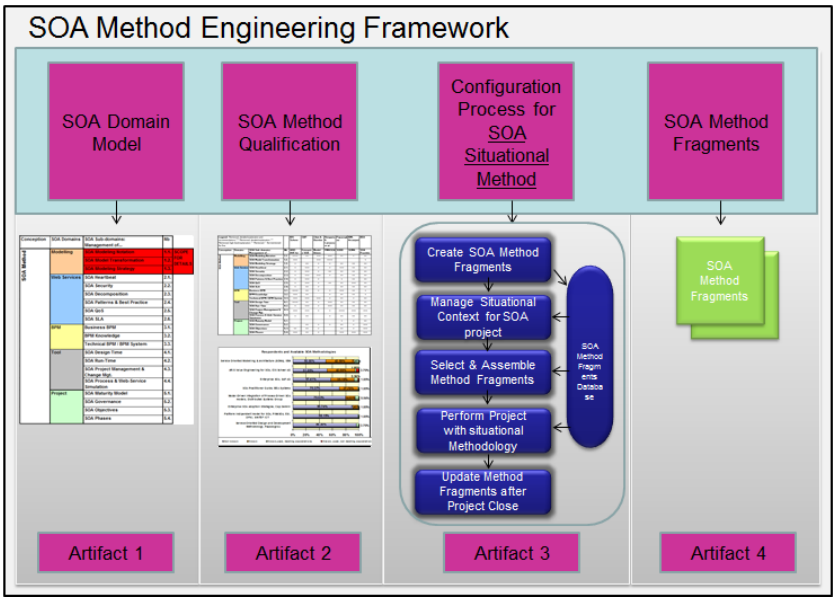



**Figure 27: Overview Chapter 3**

The value provided by resolving the posed research questions in chapter one has been detailed. Nevertheless, we will quickly remember the key value contribution areas in brief:

1. The SOA Domain Model (Artifact1) summarizes sub-domains required for SOA implementation. This SOA Domain Model enables the qualification of existing SOA Methods and finally also directions on method capabilities (Artifact 2).

2. A configuration process for SOA Situational Method is created (Artifact 3) assuring a situational application. The risk of non-fitting SOA method or method application failure should be significantly reduced. Furthermore, it is an accepted engineering method specifically for methods. Finally, method fragments are created from existing SOA Methods (Artifact 4) to demonstrate the formalization of method content using method engineering principles.

3. The selected viewpoint of top-down, model-driven and process-orientation allows academia and industry as well to select more efficiently candidate modeling languages and integration strategies.

## 3.1. Artefact 1: SOA Domain Model Details

### 3.1.1. Grouping of SOA relevant domains

As the term “domain” and “conception” has been defined in section 2.1.2., we consider “conception” as the SOA Method and the “domains” as any subsets related to this conception.

A first attempt to conceptualize the different subjects encountered through desk review and structuring them into domains has been done to visualize the state-of-the art. This is a personal view, where clusters have been created as a first attempt to structure the overwhelming number of terms, issues, criteria’s etc. If all domains around the proposed conception of SOA Method are taken together, the following landscape can be drawn in a non-formalized mind-map:

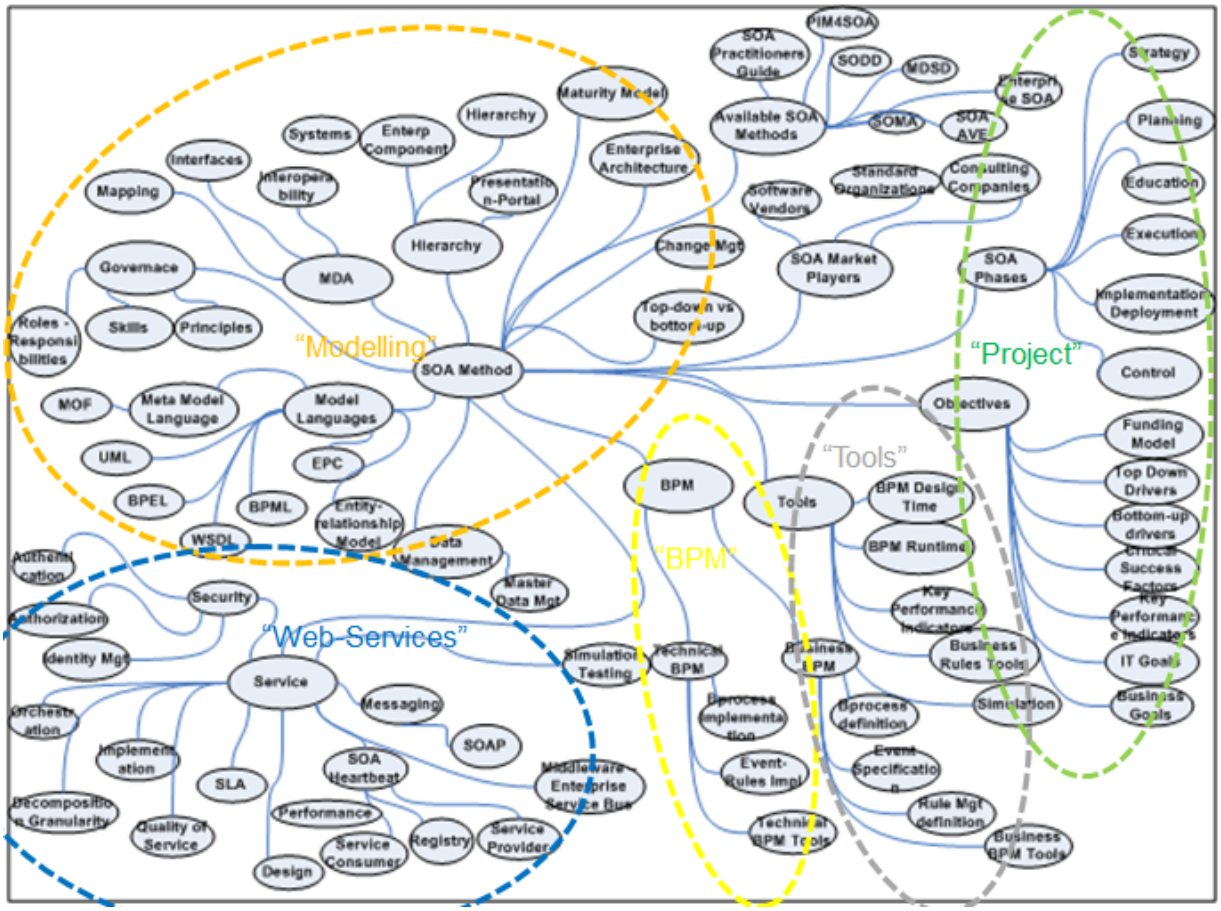



**Figure 28: SOA Method Domains**

The domains have been gathered through the analysis of the state of the art of SOA methods and the principles of applying a model-driven and process-oriented implementation approach. Figure 28 enumerates domains and components relative to the proposed conception of “SOA Method”.

The formalized SOA Domain Model presented in the next sections is structured into a table format, which is easier to oversee and exploit. Also sub-domains have been created to facilitate the structure of the model. These sub-domains are sub-groups within the domains.

They have been structured based on topics without considering sharp criteria's to distinguish or define the borderlines. Also as only specific viewpoints have been selected, these domains and sub-domains are not intending to guarantee exhaustivity. The chosen decomposition was done based on the state-of-the-art SOA Method content. Furthermore, only the modelling domain will exemplarily be used to formalize method fragments. The SOA Domain is including SOA Sub-Domain as follows:

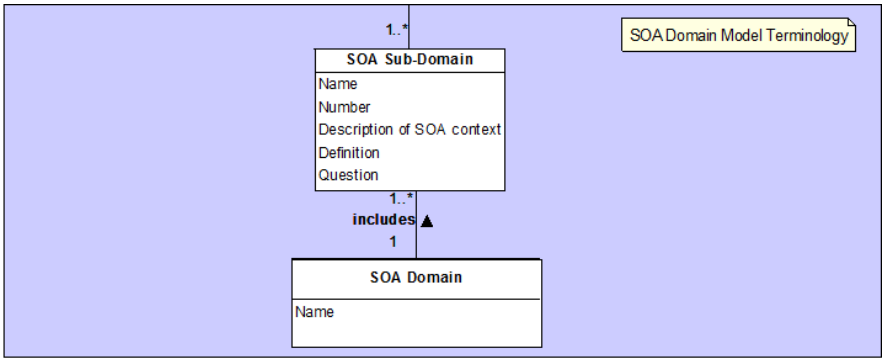



**Figure 29: SOA Domain and SOA Sub-Domain Terminology**

**Table 19: Attributes of SOA Domain Class**

| Class | Attribute | Description |
|---|---|---|
| SOA Domain | Name | One of the 5 SOA Domains (SOA Modeling, SOA Web-Services, SOA BPM, SOA Tool, SOA Project) |

The SOA Domain Model is an artifact from this thesis which includes 5 different SOA domains. The respective SOA domain is including sub-domains identified in the state-of-the-art and described in detail in section 3.1.

**Table 20: Attributes of SOA Sub-Domain Class**

| Class | Attribute | Description |
|---|---|---|
| SOA Sub-Domain | Name | Name of Sub-Domain |
| SOA Sub-Domain | Number | Number of Sub-Domain |
| SOA Sub-Domain | Definition | Definition of Sub-Domain |
| SOA Sub-Domain | Description of SOA context | Clarification of Sub-Domain in the context of SOA to ensure good understanding of decision to be taken |
| SOA Sub-Domain | Question | Question derived from Sub-Domain to be answered/decided on. |

The SOA sub-domains are further described by a unique number, definition and the context to be applied for the SOA situation. The sub-domain is defined in general and then a description with SOA context is given. Furthermore, questions to be resolved are explained.

Concrete examples will be explained in section 4.2.1.

### 3.1.2. Introduction to SOA Domain Model

This SOA Domain Model (table 21) is a summary of the state-of-the-art for a process-oriented and model-driven SOA Implementation. Each of these 5 domains is including sub-domains in the context of a SOA implementation project:

**Table 21: SOA Domain Model**

| Conception | SOA Domains | SOA Sub-domains: Management of... | Nb | |
|---|---|---|---|---|
| SOA Method | Modelling | SOA Modeling Notation | 1.1. | SCOPE FOR DETAILS |
| | | SOA Model Transformation | 1.2. | |
| | | SOA Modeling Strategy | 1.3. | |
| | Web Services | SOA Heartbeat | 2.1. | |
| | | SOA Security | 2.2. | |
| | | SOA Decomposition | 2.3. | |
| | | SOA Patterns & Best Practice | 2.4. | |
| | | SOA QoS | 2.5. | |
| | | SOA SLA | 2.6. | |
| | BPM | Business BPM | 3.1. | |
| | | BPM Knowledge | 3.2. | |
| | | Technical BPM / BPM System | 3.3. | |
| | Tool | SOA Design Time | 4.1. | |
| | | SOA Run-Time | 4.2. | |
| | | SOA Project Management & Change Mgt. | 4.3. | |
| | | SOA Process & Web-Service Simulation | 4.4. | |
| | Project | SOA Maturity Model | 5.1. | |
| | | SOA Governance | 5.2. | |
| | | SOA Objectives | 5.3. | |
| | | SOA Phases | 5.4. | |

Each sub-domain is rapidly defined to ensure a common understanding and is then explained in the context of SOA method application. As already emphasized in the introduction, only the domain “Modelling” will be detailed and method fragments created. The interrelationship between sub-domains and deeper layers (here: activities) will be introduced and explained in detail in table 22.

## 3.1.3. SOA Domain Modelling

The SOA domain "modelling" is grouping all sub-domains relative to models created and used in SOA engineering.

### 3.1.3.1. Sub-domain 1.1. SOA Modeling Notation

The issue to resolve in the context of SOA is to select the best suited modelling languages for representing ideal candidate notations to use for SOA implementation. On each level of abstraction, different models (refer to definition 19) are available and need to be evaluated for the best path to follow. Some modelling notations are more suited or used than others. This has been explained in detail in section 2.2.3. Between strategy and process layer, the issue of bridging between models is essential, whereas for process and IT layer the specific characteristics of model language (process language and implementation language) becomes more important (section 2.4.). The notations on the strategic abstraction layer can hardly adhere to all of the three criteria's (syntax, semantics and automation) as automation is very difficult to achieve. Business Rules, Events and organizational information are an important part of (process) modelling notations as they indicate when activities are triggered, by whom and how exactly specific rules need to be applied. Method fragments will later in the method application create artefacts, which are instances of e.g. EPC or BPMN models which are resolving a specific issue on each level of abstraction. These artefacts are then called "product fragments".

Directly related to modelling notation is the underlying issue of Meta Models. We will not consider meta models as a separate issue item as they are differentiation criteria of various models. A meta-model does not in general describe a process but the abstract syntax of a language. It is a model of models (expressed in one language) and as such describes all possible models expressible in that language (=syntax). Furthermore, every meta model is based upon another meta model whereas flexibility and reuse are important criteria's for the modelling languages to be used on a high level of abstraction [Gru93][Gru04].

Meta models become important in case of translation or mapping of languages to other model languages on same abstraction level (e.g. UML class diagram, BPMN, EPC) or between different abstraction levels (business process notations vs. implementation notations). Modeling notations relying on meta models can help to translate these notations easier because they are more formalized than languages without. On the other hand, notations without meta-model are often also well described through conventions (objects to use, symbols to use, attributes to use, connections to use, how to model etc.) and specific mappings or translation mechanisms are proposed in academia. UML class diagram with MOF is the most known and used meta- model. EPC and BPMN have meanwhile also reasonably formalized meta-models, which allow also better interfacing to other models related to the model transformation problem.

Mata data is an important issue to resolve as web-service will need to create, read, update and delete metadata (CRUD) in the repository. Therefore, metadata needs to be under control through data governance and reliable tools. Data as input or output for a business service is crucial to describe in processes (e.g. EPC, BPMN, UML-class etc.). Data models can help to repertorize data and to show relationships between each other (e.g. UML-class diagram, ERM etc.) A specific meta-model for metadata has been developed by the OMG and is called

Common Warehouse met model Specification (CWM) [IEEE00]. These models become important for applying MDA as formal models need to be transformed by generators.

#### 3.1.3.2. Sub-domain 1.2.: Model Transformation

In order to be as efficient as possible with the effort to conceptualize SOA strategy and business processes and then to translate business requirements into web-service description and implementation, the question of how to interface the different kinds of models is crucial. If MDA principles are applied, the transformation (refer to definition 30) between models can become real. Therefore, the notation itself is important (relying on meta model) but also the tool allowing a transformation between models. The automation of information exchange and the so called "round trips" is in an automated way so far only possible on the deepest level of detail within the UML family of notations and tools. MDA principles do claim to translate formal machine readable process and data models to code via code generators. The investment in MDA can pay off since the PIM and PSM abstraction levels are addressed. By MDA principles, a huge issue discussed in academic and industrial world known as the "exchange problem of semantic information" between layers is addressed. Code generators are usually used to understand the semantics described in modelling languages to translate into code (XML, WSDL, UDDI, SOAP) [IEEE00].

#### 3.1.3.3. Sub-domain 1.3.: Modeling Strategy for SOA Delivery

Projects managers and architects need to carefully evaluate the different approaches (refer to definition 6) such as e.g. top-down, meet-in-the-middle, bottom-up upfront to ensure a successful SOA implementation (in time, in budget, in required quality) meaning to spend the optimal amount of resources. Terlouw et al [TTJ09] have defined a so called "Delivery Strategy Assessment Method" (DSAM) determining the most appropriate SOA delivery strategy for an organization as introduced in chapter one.

Each SOA Domain can be further detailed with generic activities to be performed and related artefacts. The term "activity" has been introduced in section 2.6.3. Basically, the term is used for generic activities, which would be required for a SOA engineering method. These activities have been gathered through the state-of-the-art on SOA Modelling. The related artefacts are indicating what outcome could be expected. Mostly, it is a model, which is an outcome of this specific activity. This will be fine-grained and further explained in in table 15 which is formalizing the identified sub-domains into "activities".

In order to overcome the issue of specific method fragment semantics, an important term of "activity" is introduced.

This activity corresponds to a generic project activity to be performed in a specific context and is the bridge between the method fragments and the SOA Domain sub-domains. For example the SOA Domain "Modelling" will include a sub-domain "SOA Modelling Notation". This sub-domain is addressing different activities such as "Create Strategy Model", "Create CIM Model", "Create PIM Model" etc. For each of these activities, one or more method fragments with the semantic of fragment providers will be available for selection. This will be described in detail in chapter 4.2.1.

We are not claiming exhaustivity as this is simply not possible, but it is a rather complete summary of relevant activities towards the chosen viewpoints:

**Table 22: SOA Domain Modelling Details**

| Domain | Sub-Domain | Activitiy |
|---|---|---|
| Modelling | 1.1. SOA Modeling Notation | Model Strategy with BSC |
| | | Model Strategy with BMM |
| | | Model Strategy with e3forces |
| | | Model Strategy with e3value |
| | | Model Strategy Key Performance Indicators |
| | | Model Startegic Goals, I* |
| | | Model Value Chain for Process Overview |
| | | Model Business Requirements with EPC |
| | | Model Business Requirements with BPMN |
| | | Model Business Requirements with IDEF |
| | | Model Business Requirements with PetriNet |
| | | Model Business Requirements with UML |
| | | Model Data Glossary with UML |
| | | Model Data Glossary with ERM |
| | | Map Services and Data with Entity-Relationship-Diagram |
| | | Model Technical BPMN |
| | | Model Technical BPEL |
| | | Model Technical YAWL |
| | 1.2. SOA Model Transformation | Model Strategy2CIM e3Value2BPMN |
| | | Model EPC2BPEL |
| | | Model BPMN2BPEL |
| | | Model UML2BPEL |
| | | Model EPC2BPMN |
| | | Model EPC2UML |
| | | Model BPEL2WSDL |
| | | Model UML2WSDL |
| | 1.3. SOA Modelling Strategy | Select SOA Modelling/Implementation Strategy Top-Down |
| | | Select SOA Modelling/Implementation Strategy Meet-In-The_Middle |
| | | Select SOA Modelling/Implementation Strategy Bottom-Up |

As this thesis is focussing on the SOA Domain Modelling, only the activities and artefacts of this domain are enumerated in table 22. These activities are finally producing artefacts, in this case different type of models based on different notations. It could be well imaginable to detail similarly to the Modelling domain also other domains in future works (Chapter 7).

## 3.1.4. SOA Domain Web-Services

The SOA domain web-services is grouping all sub-domains relative to web-service aspects. This domain will not be detailed and applied to SOA fragments.

### 3.1.4.1. Sub-domain 2.1.: SOA Heartbeat

The issues are more technical nature to ensure a proper functioning between the three actors with an ESB integration between Service provider and Service consumer. For the definition, pls. refer to definition 5 in the introduction.

### 3.1.4.2. Sub-domain 2.2.: SOA Security

According to Peterson, SOA security ensures full enforcement of authentication, authorization and identity management policies [Pet08].

The issue here is to decide about the level of security that seems to be appropriate for each service, the related processes and the underlying infrastructure [Bue07]. The security need of a home banking service is different to the service requirement for weather information on Google.

#### 3.1.4.3. Sub-domain 2.3.: SOA Decomposition

The foundation of SOA decomposition (=decomposition is a well-established technique) is an enterprise business model, which contains the primary representation of the resources and processes for meeting operational, tactical, and strategic business goals. A business model is an essential component of a successful service-oriented decomposition, ensuring consistency and flexibility of resulting services across the organization (motivations for adopting an SOA approach).

There are many approaches to defining enterprise business architecture. Some architects base their interpretation of business architecture on a corporate organization. A business function or process model can be used as a starting point to draw connecting lines between vertical functions and horizontal processes to describe a cross-functional process within the business.

The key component of a business model is the description of the enterprise business processes, which defines their supporting activities, inputs, and outputs. Process activities provide the foundation for defining the enterprise services. Using the business model as the starting point in decomposing the solution into services facilitates the alignment between business and technology—one of the objectives of the SOA approach. Decomposition is a well-established technique. Depending on the objective, many decomposition criteria can be applied. The decomposition criteria have a significant impact on architecture goals such as performance, flexibility, comprehensibility, development time, changeability and reuse [Lub07].

#### 3.1.4.4. Sub-domain 2.4.: SOA Patterns & Best Practice

SOA patterns & best practice are preconfigured processes and embedded web-services to import into the BPM design tool or the BPM run-time tool.

Patterns and best practice can help to realize efficiency gains, meaning not to invent everything from scratch. The issue related to this is that these preconfigured proposals are rather generic and not necessarily on the right level of detail than what is expected or needed.

#### 3.1.4.5. Sub-domain 2.5.: Quality of Web-Service (QoS)

QoS for web-services is defining the level of real time availability and performance metrics and is measured through a service-level agreement. A quality system is supporting deployment, configuration, versioning, monitoring, management and auditing of web-services.

To achieve the QoS defined by the business, each service endpoint should be managed as a resource. This includes the invocation of services (service consumer) as well as the

application functionality exposed as a service (service provider). When down, the management tooling should provide a means of troubleshooting, and, better still, a method of monitoring and alerting of issues before failure.

#### 3.1.4.6. Sub-domain 2.6.: SOA Service Level Agreement (SLA)

SLA is a contract or product fragment which describes in detail the expected levels of services (for web-services e.g. performance, downtime, etc.) or business requirements in order to a.) improve service levels and is b.) used as baseline for measuring the service performance (e.g. metrics like processing time, messages per hour, rejected transaction counts and queries per day) between service provider and service consumer.

Agents can be deployed to gather the desired metrics, and code can be added to applications to process these metrics and behave accordingly. In practice, this is rather difficult, because service end points may be added or changed. New services might be offered or existing service levels need to be improved. Therefore, a process needs to be implemented in order to enforce policies (per message basis, policy compliance verification etc.) [CH06].

### 3.1.5. SOA Domain BPM

The SOA domain ”BPM” groups all sub-domains relative to process management aspects. This domain will not be detailed and applied to SOA fragments.

#### 3.1.5.1. Sub-domain 3.1.: Business BPM

Brocke and Rosemann [BR10a] in their Handbook of BPM, the management of this approach is focusing on aligning all aspects of an organization with the wants and needs of clients. It is a holistic management approach that promotes business effectiveness and efficiency while striving for innovation, flexibility, and integration with technology. Business process management attempts to improve processes continuously. Supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information [vdAtH05]. It could therefore be described as a "process optimization process." In general, BPM will enable organizations to do “the right things right” meaning to be effective and efficient.

BPM is materializing in business process modelling, which is important to a process-driven SOA implementation. Without the knowledge on activity sequences, the triggers and rules to these activities, it is difficult to imagine an efficient approach for identifying activities to be supported by web-services. Finally, SOA is seeking to improve the business processes by using web-service technology. The set-up of processes with a notation of choice is a prerequisite for process-driven SOA. According to Klueckmann [Klu07] it is not feasible to have a real working SOA, if the processes are not known. As processes are defined and executed by business users, their involvement is key. Therefore, “the SOA implementation method needs to be business-driven BPM because it is the actual business processes and not the orchestrated services that determine SOA design.” Leading software providers such as,

ORACLE, Fujitsu, SAP, Tibco, Software AG and IBM recognized this fact and integrated in their application suites now also process modeling tools to represent business requirements. Klueckmann resumes that all these vendors included BPM into their SOA implementation approach as they all recognized that the pure technical approach failed.

#### 3.1.5.2. Sub-domain 3.2. Business Process Knowledge Mgt.

Business process knowledge is the summarized information about business process content in order to efficiently improve the process by applying web-service technology. This can materialize in guidelines, procedures, databases, human knowledge etc. The knowledge and experiences need to be made available to be exploited in an efficient manner.

Without knowledge on processes it is simply unrealistic to efficiently improve the process. An understanding of who is doing what, with what application and data with what objective and result is needed to define web-services on the right granularity level to increase process efficiency. This knowledge is important for SOA Method usage.

#### 3.1.5.3. Sub-domain 3.3. Technical BPM or BPM System

A generic software system that is driven by explicit process designs to enact and manage operational business processes. The system should be process-aware and generic in the sense that it is possible to modify the processes it supports. The process designs are often graphical and the focus is on structured processes that need to handle many cases [AHW03].
For SOA implementation, an important task is to find the appropriate tool to execute process and to enhance the process with needed web-service descriptions, orchestration information and business rules execution. The platform independent models need to be exported into the technical environment. Once web-services are deployed, the technical BPM system needs to provide testing functionalities as well as performance monitoring as input for SLA measurement.

### 3.1.6. Domain SOA Tool

The SOA domain "Tool" is grouping all sub-domains relative to useful tooling for SOA implementation. This domain will not be detailed and applied to SOA fragments.

#### 3.1.6.1. Sub-domain 4.1.: SOA BPM Design Time

A BPM design time tool used in the SOA context is providing the design facilities to model different types of models on different abstraction levels in order to define business services and to translate them into web-services.

Tooling capabilities are essential to cope with requirements of process-oriented modelling of SOA. Different notations need to be available (strategy model, business requirement model, technical model, web-service description).

#### 3.1.6.2. Sub-domain 4.2. SOA BPM Run-Time

A BPM run-time tool for SOA is providing the tool support to implement, test, execute, simulate and control processes including the related web-services.

The integration of design tool and run-time tool is key. Interfacing models from design layer into execution layer should be possible. The functionalities need to support the implementation, testing/simulation, execution and control of processes including web-services and business rules.

#### 3.1.6.3. Sub-domain 4.3. SOA Project Mgt & Change Mgt Tool

A project management tool is supporting the project manager with planning organizing and managing resources of project (activities, time, cost, dependencies) and execution of project (Status on progress) to meet specific SOA project goals and objectives.

A change management tool is supporting the project manager to ensure all changes are assessed, approved, implemented and reviewed in a controlled manner.

The project manager needs to be supported by a flexible tool allowing him the construction of a situation-specific project method depending on requirements of the organisation such as the guidance on the different issues mentioned through the SOA Modelling Domain and the Web-Service Domain. Project manager needs also this tool for communication purpose in the organization such as e.g. evangelization about the SOA topic, training preparation and to communicate between the various different types of profiles needed for this specific type of project.

#### 3.1.6.4. Sub-domain 4.4. SOA Process & Web-Service Simulation and Testing

Process & web-service simulation tools is about providing a system to support assessment, control and testing processes and web-services related to business requirement fulfilment (functional testing) within an acceptable performance (technical testing).

The system functionalities needed should be integrated into the BPM & Web-service run time environment. Simulations can be very helpful and value added. The extent needs to be decided on. The risks to mitigate are important: If web-services are released without thorough testing, showing malfunctions or non-performance, business users or clients will be upset and business support for the project will be seriously damaged.

## 3.1.7. Domain SOA Project

The SOA domain "Project" is grouping all sub-domains relative to project management aspects. This domain being will not be detailed and applied to SOA fragments.

### 3.1.7.1. Sub-domain 5.1. SOA Maturity Model

According to BPtrends [IA07], a SOA maturity model is used to assess the current state of SOA adoption of an organization. The model is used as a yardstick to take stock of AS-IS state and develop a transition plan to lead the enterprise to the TO-BE state. The ultimate aim would be to achieve optimized business services that can nimbly adapt to changing business scenarios.

The issues to clarify are related to the decisions on scope of SOA adoption (Department, Business Division, Cross Division, Enterprise Wide), SOA Maturity Level (capabilities of the architecture e.g. initial services, architected services, business services, measured services, optimized services). Furthermore, it needs to be decided what progress is planned or to be achieved on timescale e.g. different stages to achieve. This is linked to culture, budget, risk appetite etc. Buckow et al [BGPPW+10] have analysed and evaluated available SOA Maturity Frameworks (ACMM, Inaganti/Aravamudam, Sonic, Oracle). Based on the SOA Maturity Framework reviews, they have also developed a framework to analyse SOA abilities of available standard platforms on the market.

### 3.1.7.2. Sub-domain 5.2. SOA Governance

Leusse, Dimitrakos and Brossard [LDB09] are defining the SOA Governance as "processes with roles & responsibilities used to oversee and control the adoption and implementation of SOA. This governance is using recognized practices, principles and government regulations in order to provide optimum service quality, consistency, predictability and performance of web-services through the application of policies."

There are two areas to address: a.) the governance process of web-service implementation during the project and b.) the set-up of the service level agreement with the objective to control, measure and improve web-service. Within the governance set-up, roles & responsibilities need to be created, control processes have to be implemented and finally to be assessed during the implementation. On-going measurement of web-services is also an important process. The SOA heartbeat as introduced in chapter 1 with registry, service consumer, service provider needs to be managed based on efficient policies. Furthermore, the costing model needs to be measured based on the SLA.

### 3.1.7.3. Sub-domain 5.3. SOA Objectives

SOA objectives are targets to be achieved to reach the quantified benefits or benefits that are not possible to quantify for the SOA projects. SOA critical success factors enable the easier achievement of these objectives. The SOA Return of Investment (ROI) is the quantification of benefits in relation to the overall cost.

SOA Objectives, SOA, Critical Success Factors (CSF), SOA benefits, SOA ROI etc. are all elements which should be detailed in the product fragment "project brief/summary". For this, it is necessary to have an experienced SOA project manager for a consistent, realistic and complete document.

#### 3.1.7.4. Sub-domain 5.4. SOA Phases

A SOA phase is a group of activities to facilitate project management.

The provided approach in this thesis will facilitate this task to identify the relevant process fragments and related product fragments following the principles of the method engineering approach. This is important as we will later re-use the term "SOA phase" within the method engineering method. The SOA phase decomposition for the application here is done on 9 generic phases. These phases are:

1.) SOA Strategy
Activities within SOA Strategy are e.g. definition of SOA targets, requirements, scope, impact analysis, IT integration and technology used, data used, capabilities of the system etc.

2.) SOA Planning
Activities within SOA planning are e.g. definition and planning of resources, creation of situational SOA method and project plan etc. etc.

3.) SOA Education
Activities within SOA education are including e.g. the identification of knowledge gaps per role involved in the project and addressing by training etc.

4.) SOA Specification
Activities within SOA specification are e.g. including Modelling on Strategy Layer (e.g. BSC, BMM, Strategic Modelling), Modelling on CIM layer (EPC, BPMN, UML), Modelling on PIM layer (UML, BPEL), Modelling on PSM layer (Web-service descriptions, Code) etc.

5.) SOA Design
Activities within SOA design are including e.g. service design, service granularity, service decomposition, detailed SOA heartbeat design etc. etc.

6.) SOA Development & Implementation
Activities within SOA development & implementation include e.g. the definition of web-service interfaces and service implementation descriptions, implementation of SOA heartbeat, functional/technical testing, roll-out of web-services etc.

7.) SOA Control
Activities within SOA Control are e.g. the implementation of defined roles & responsibilities enforcing web-service policies, web-service performance monitoring, KPI measurement and monitoring and control of SLA etc.

8.) SOA Change Management

Activities within SOA change mgt. are e.g. stakeholder analysis, change mgt. plan, information, communication, training, change process for web-services etc.

9.) SOA Governance
Activities within SOA governance are the definition of registry and related policies including testing and auditing.

### 3.1.8. Summary on SOA Domain Model

The introduced SOA Domain Model is a reference model that is describing domains and sub-domains as defined in the introduction under the premise of a process-oriented and model-driven architecture. These domains structure or cluster a group of topics, which are relevant when SOA is implemented. For sub-domains very precise activities can be defined, which are producing artefacts. These activities can be done, but at this state, there is no indication in what context these activities apply and what conditions are related to it.

## 3.2. Artifact 2: Qualification of SOA Methods with SOA Domain Model

The outcome of this section will show how the SOA Domain model will serve to evaluate existing methods on domain completeness and underline the initial research need of defining a SOA engineering method. There is none of the available SOA Methods using ME principles

### 3.2.1. Qualification of available SOA Methods with SOA Domain Model: Selection of relevant Methods

The exhaustive analysis and evaluation of all existing methods and pieces of information would take too long fearing the added value would be limited for the huge time-investment needed. This does not mean that pieces of information towards specific topics are not valuable, but the objective in this section consists in showing how the SOA Domain Model can be applied on “SOA implementation methods” and that available proposals are not covering the complete span of the SOA Domain Model. Also, some industry-specific SOA methods exist e.g. for education [BGLOS+09].

In order to select relevant methods for the detailed analysis, some need to be discarded as they do not fit within the proposed definition of method. The definition of method is around phasing and grouping activities in a plan, to use modelling to abstract from the very complex reality related to a specific chosen viewpoint [ISO06a] and to recognize the necessity of tools to work efficiently and to cope with complexity. As defined by Kruchten [Kru95], he introduced the definitions of “conception” and “domains”. In his terminology, “conception” corresponds to the SOA Method and the “domains” correspond to “any coherent subsets of issues related to this conception.” This is the reason why software engineering methods were not included into the analysis as they have not SOA as a conception.

To increase justification, we also refer to the method definition of Vernadat [Ver92]. It needs to be "(1) a set of methods, (2) models and (3) tools to be used in a structured way to solve a problem."

To justify the choice, table 23 is summarizing the methods in a table evaluating if the three defined criteria are met and - finally cope with the proposed definition:

> First, by a set of methods, we understand a set of series of activities grouped in a phase to be performed in a sequential order.
>
> Second, referring to the definition of model (Definition 19), we understand that a model is "a purposely abstracted and unambiguous conception of a domain." The domain in this case corresponds to the purpose of SOA implementation and therefore, the model as we understand it here needs to fit into the SOA purpose.
>
> Third, by tools we understand the possibility to use any software tool helping to structure, formalize, re-use and facilitate methods and models. It should be mentioned how a tool can contribute to this.

**Table 23: Selection criteria's for current SOA Methods**

| Name of SOA Method | Phasing/grouping activities in a plan | Modelling, notations for the SOA domain | Tool usage, contribution, requirements | Comments |
|---|---|---|---|---|
| Architecting Industry Standards for Service Orientation [Lee05] | | | X | Very technical driven on SOA web-services. Viewpoint of Microsoft technology, Tool suite Microsoft |
| ARIS Value Engineering for SOA [Yva06] | X | X | X | ARIS Value Engineering is steps with phasing, modelling choices given (BSC, EPC, BPEL, WSDL), strong explanation how to use BPM design tools, Run-time missing. |
| CBDI-SEA SOA Reference Framework [But09] | (X) (Only SOA Lifecycle) | X | | More comprehensive framework than implementation methodology. Similar to BEAs practitioner guide but without phased approach. Is more focussing on SOA lifecycle "manage, consume, provide and enable". Modelling and notations is summarized in "typical format" which describes how key artefacts (in this case product artefacts) can be formalized. No tooling information available. |
| Enterprise SOA [WM06] and update with Accelerated Transformation to SOA [SAP08] | X | X | X | Full SAP method with phased approach, modelling notation (BPMN 2.0), and SAP tools such as Process Integrator & Configuration Environment. |
| Enterprise SOA Adoption Strategies [Jon05] | X | | | Pure functional & consulting-driven top-down phased approach to decompose business functions into web-service candidates. Neither modelling nor tooling details. |
| Model-Driven Integration of Process driven SOA Models [ZD06] and View-based Integration of Process-Driven SOA Models AT Various Abstraction Levels [TZD08] | X | X | X | View-based research approach based on modelling (only UML) using a translation tool (with DSL FRAG) from UML into web-service description. |
| Platform-independent model for service-oriented architecture [BL06] | X | X | X | Phasing very light, specific modelling language using MDA levels and Eclipse tooling using transformation mechanisms. |
| Service-oriented Design and Development Method [PvdH06] | X | X | X | Clear phasing with explanation, modelling and light tooling requirements explanations. Focus |

| | | | | |
|---|---|---|---|---|
| and [NvdHP+09]. | | | | on business service identification. |
| Service oriented Modeling & Architecture [Ars04] and [AGAAG+08] | X | X | X | Clear phasing with explanation, light modelling explanation and focus on IBM tools and software. |
| ORACLE Unified Method for SOA [ORAC11] | X | X | X | Comprehensive SOA framework including phased activities, high level modelling notation explanation and tooling requirements based on ORACLE 11g platform and other ORACLE products (e.g. jDeveloper) |
| SOA Reference Model for Service Oriented Architecture [MacK06] | | X | | More a reference model, no phased activities nor tooling requirements |
| SOA Adoption Model [GART06] | X | | | high-level phased approach for management with viewpoint from SOA maturity analysis and best practice. Neither modelling nor tooling. |
| SOA Delivering Strategies [NL05] | X | X | | Phased approach light modelling description, no tooling. |
| SOA Organizational Roadmap [KBS06] | X | X | | Phased approach light modelling description, no tooling. |

Therefore, the following SOA implementation methods from the analysed list meet the 3 criteria's:

- ARIS Value Engineering for SOA (AVE for SOA)
- Enterprise SOA
- Model-Driven Integration of Process driven SOA Models / VbMF
- Platform-independent model for service-oriented architecture (PIM4SOA)
- Service-oriented Design and Development Method(SoDD)
- Service oriented Modelling & Architecture (SoMA)
- ORACLE Unified Method for SOA

These candidates have been carefully analysed along the domains of the SOA Domain Model with the objective to compare them in a structured way and to evaluate on which dimension the methods are strong or weak meaning if and how activities mentioned earlier as subset of criteria's in specific domains are addressed or not. The criteria's are further detailed by activities as illustrated in table 22. The activities are a rather complete view of the sub-domains. The completeness of a method for a particular sub-domain is not dependent of addressing all activities rather than proposing a consistent and logical view/approach/explanation.

Therefore, a 4 level nominal scale has been applied:

A sub-domain was not covered at all, meaning that the method is containing no sign or evidence that this topic has been addressed or considered (= No Star). Second, the sub-domain was partially covered. This includes an explanation on what it is about but only on high-level of detailand no advice is given on how to solve it (= 1 star). Third, the sub-domain was largely covered (= 2 stars) meaning that the sub-domain was explained in detail.If a sub-domain was alomost fully covered (= 3 stars) meaning the sub-domain was almost fully covered with also an explanation or recommendation on possible solutions.

In order to mitigate risks of subjective comparison, the marks have been applied in the most objective manner as possible.

**Table 24: Criteria for SOA Method Comparison**

| Analysis criteria for SOA Domain Model Issue | Mark for nominal Scale |
|---|---|
| Not Covered | No star |
| Partially Covered | * |
| Largely Covered | ** |
| Almost Fully Covered | *** |

The sections 3.2.2. to 3.2.8. are providing a condensed description on the evaluation of each existing SOA Method, which is summarized at the end in an overall comparison table. For space reasons, the details per method can be found in the Annex B. However, the first two methods are detailed on the SOA Domain "Modeling" sub-domains which are 1.1. SOA Modeling Notation, 1.2. SOA Model Transformation, 1.3. SOA Modeling Strategy. These two methods will be used to exemplarily show how method fragments can be formalized (section 4.2.) and applied in case studies (chapter 6).

## 3.2.2. AVE for SOA

The method from IDS Scheer (Software AG) is based on their companies' roots, which is BPM with an academic background. The developed ARIS Toolset was the first application in the early eighties being able to introduce different views and modelling languages. The method AVE for SOA is derived from the method AVE for BPM with the different application scenarios.

The method is strong for the modelling part, as different model types (+150) depending on viewpoints is integrated in one tool. A weakness consists of giving just high-level advice on how all these models fit together from an SOA perspective. As the SOA Method is directly related the state-of-the-art modelling tool [GAR11],which is positioned in the leading quadrant of GARTNER's market analysis, the domains "Modeling" and "BPM" are very well developed (see appendix B for details). Unfortunately, important topics e.g. the link to MDA levels is not made neither a proposition of Enterprise Architecture with conceptual levels. BPM component is also rather strong but only for knowledge and business BPM. The technical side is left to vendors with tools for BPM Run-time without explanation. Also, on the tool side, Design Time is very strongly elaborated, whereas Run-Time and SOA Performance & Simulation are weak areas. An integration with Software AGs webMethods execution engine is planned but not mature yet. The domains "SOA project" and "Web-Service" are also weak points of this method because only SOA phasing within a project and some issues about SOA governance are explained.

Related to the SOA Domain Modelling, the following criteria's have been rated as follows:

Sub-Domain 1.1. "SOA Modeling Notation": 3 Stars (almost fully covered):
The method explains in detail through a phased approach which modelling notations are available on the strategy level (Balanced ScoreCard), on the CIM-Level (Value Chain, EPC, BPMN, UML), on the PIM Level (BPEL) and how to integrate web-service descriptions into these processes. The method of IDS Scheer is explaining in detail which of these models to use and also how to use them. The method is referring to training documentation and convention documents on how to use the relevant objects for SOA-oriented modelling. The conventions are tailored to a specific CIM2PIM (EPC2BPEL) transformation. Other

transformation mechanisms are not available. AVE for SOA is also explaining which roles should perform which modelling task and what contribution is expected.

Sub-Domain 1.2. “SOA Model Transformation”: 2 Stars (Largely Covered):
The SOA Engineering Method is based on the SOA Architect capabilities, which is offering a CIM2PIM (EPC2BPEL) transformation mechanism. Therefore, the SOA Method recommends using EPC as CIM notation. Next, the method explains in detail how to model these EPC models to ensure that they can be transformed into BPEL notation. For instance, the OR operator or back loops are forbidden as these conventions are not in line with the used translation algorithm. A semantic check can detect errors and prevent translation failures.

Sub-Domain 1.3. “SOA Modeling Strategy”: 1 Star (Partially Covered)
The different modelling strategies are just mentioned high-level without being described precisely, but the AVE for SOA method is only using top-down and deploys the details through a clear top-down approach. This is reflected by the method phases to go through which are SOA Strategy, SOA Design, SOA Implementation and SOA Controlling.

### 3.2.3. Enterprise SOA

This method is focussed on the ERP (SAP) environment. The method explains how web-services in an architecture that is driven by SAP technology can be implemented. This is the reason why the domains “modelling” and “BPM” are much weaker than the domains “Web-Service”, “Tool” and “SOA-Project”. The explanation around Enterprise Architecture with its different layers is a big strength of the method. Again, here the method is related to SAPs technology vision around their service enabled platform XI with SAPNetWaever. In the update of the method [SAP08] TOGAF is recommended for capturing Enterprise Architecture. The Configuration Environment (CE) allows a process-oriented approach starting with BPMN notation. This BPMN modelling is done in the “design-time” environment and allows deploying and executing these models directly in SAP run-time engines. This is done through the Configuration Environment (CE). The Process Integration (PI) module enables the shift from pure technology view towards a more business oriented process view. A second strength is the procedural model for service discovery. The strategic layer is not formalized in models. All other domains & issues are also not well developed.

Related to the SOA Domain Modelling, the following criteria’s have been rated as follows:

Sub-Domain 1.1. “SOA Modeling Notation”: 1 Stars (Partially Covered):
SAP Method is only referring to BPMN notation in the Configuration Environment (CE). It is build-into the technical integration, meaning that BPMN is executable. The method explains not how exactly BPMN needs to be modelled as they are referring to training services given at SAP. There is no strategic model or business requirement model. The proposed BPMN style is technical and integrating business objects and transactions using web-services.

Sub-Domain 1.2. “SOA Model Transformation”: 0 Stars (Not Covered):
The SAP method is not proposing any integration, as the approach is starting with a technical BPMN model. If the technical BPMN notation conventions are respected following SAPs guidelines, the models can be executed in the operational run-time.

Sub-Domain 1.3. “SOA Modeling Strategy”: 1 Star (Partially Covered):
The method is explaining the existence of different options, but this is not further detailed.

### 3.2.4. Model Driven Integration of Process Driven SOA Models

Zdun & Dustdar describe in their method based on model-driven software development (MDSD) an academic approach with UML2, domain specific languages (DSL), a meta-meta model (MOF) and the own developed pattern language "Frag" for the syntax. Both are recognized computer science researchers with a more technical background. As the method starts on the technical PIM-level in MDA, Modelling & Interfacing within the "Modelling" cluster is well addressed but refers to interfacing technical UML2 models. Therefore, the technical design part is well explained, whereas the Strategy, MDA, Knowledge & BPM business parts are missing because not in scope. There is some explanation about the model-driven design process (SOA phases) and how to decompose process into "Macro-Microflow" (decomposition & granularity). If this method could be enhanced by missing SOA domains, it could be a valid method for SOA architects to apply, if UML2, DSL and "Frag" [Zdu09] are the decisions related to language and syntax.

Tran, Zdun and Dustdar [TZD08] [TZD08b] describe an approach to eliminate issues related to interoperability and reusability of process models (refer to chapter 2.4.2.). They describe shortcomings in extensibility of models related to direct transformation mechanism from one model to another. The presented View-based model-driven framework is based on the view-concept and meta-models for different views such as control flow, collaboration, information, transaction and new concerns that could come up. The ultimate objective is to develop an approach for automatic translation between models by using an approach based on views, concerns, MDSD, Meta-Meta Model and Meta Models.

### 3.2.5. PIM4SOA

This is a method developed in the EU-funded project on interoperability research, ATHENA. The strengths are clearly on "modelling and Interfacing". OMGs Model Driven Architecture principles are followed and the developed tool is working in an Eclipse environment. UML 2.0. profiles have been created for POP (Process, Organization, Product) PIM4SOA and Web Services in Rational Service Modeler (RSM). Furthermore, BPEL and WSDL are used and interfaced to productive environment via standard interfaces, which is one of the strengths on the tool side, but the BPM foundation as well as the understanding of the "web service" domain is not integrated in this approach. As the name tells, "PIM4SOA" is neglecting the CIM layer.

### 3.2.6. Service Oriented Development & Design (SODD)

This method from Papazoglou and Van der Heuvel is a refined method based on IBM's SOMA Method. In SODD, shortcomings in IBMs SOMA Method have been identified and mostly closed e.g. the view on Business BPM or SOA Governance. Some Model types are stated, yet, there is no description on how they fit together and how to match them to MDA abstraction levels. Key principles as the foundation for service-based process design are well explained. The planning and design phase gives important information about scoping of processes, how processes can be identified and the different realization options. Furthermore, service design concerns are well explained and how services could be specified. Service design has been further developed and detailed with a methodology for "business service

engineering" [NvdHP+09]. The objective of this further method development is to identify candidate business services including criteria's such as functionality, scope, reuse and granularity applying a gap analysis approach. Papazoglou does not mention the MDA principles nor is the focus to give an overview of models that could be used other than to use BPMN as a candidate business process language, WSDL for services and BPEL for orchestration. UML or any other business process modelling language does not play any role whilst the method focus on the importance of process modelling, design, analysis etc. The strategy phase, strategy concepts and methods are also not taken into consideration.

### 3.2.7. SOMA

This method is focusing primarily on the domain of "web-services" as IBM comes traditionally from the technology side. IBM has been the first company in the market with a sound SOA Method. The principles are explained around the classical layers of SOA. Another strong domain is "SOA project" as phasing, approach and governance are described. The SOMA life-cycle describes a high-level flow with 22 activities over 7 different phases. Weak points are clearly the domains "Modelling" and "BPM". The best described issues in the weaker domains are the choice between "top-down" or "bottom-up" and questions around Enterprise Architecture layers for SOA. Tools from IBM used in this method e.g. Rational Rose gives advantages for issues on "Technical BPM" and "BPM Runtime", but the integration of views and different modelling types for business language is missing. Rational Rose includes a wide range of UML notations, but is missing notations on CIM layer. Following to Johnson [Joh05], IBM has adapted the RUP method for SOA into the IBM Rational product. This also illustrates the wish of IBM to propose SOA adaptations in their tools and back-ends. Arsanjani started 2004 with the development of the method, which has been updated 2008 by Arsanjani and IBM Researchers [AGAAG+08].

Based on Arsanjanis' work, Zimmerman [Zim09] an IBM researcher has enhanced the presented method. The thesis is presenting a framework which is called SOA Decision framework (SOAD) consisting of 7 steps and a re-usable architecture model (RADM). The advantage of the thesis is the deepness of the framework with nearly 389 decisions. A tool is proposed for enhancing the structure of architectural SOA decisions. The thesis is not detailing them as these decisions are protected assets harvested from IBM projects. Only one decision "Invocation Transactionality Pattern" is exemplary shown in the Annex, Table 34. The framework is not taking a specific viewpoint e.g. process-oriented and model-driven, but focuses more on technical issues and related architectural decisions such as protocols, patterns, infrastructure, service descriptions etc.

For organizations starting with SOA, the framework is complex to understand and/or to apply. Furthermore, the framework is designed in a way, where the SOA architect needs considerable experience and understanding of this domain. The solution is more intended to be proposed and guided by (IBM) consultants that are well experienced using this model. If this condition is met, then this presented framework and architecture model is an excellent approach to successfully facilitate SOA implementation considering a wide range of (mainly technical) issues to work on.

### 3.2.8. ORACLE Unified Method (OUM) for SOA 5.5.

This is the most complete among the industrial methods. However, the domains "modelling" and "BPM" could be improved. There is very little information about what model types to choose and nothing is said about how to link them. The "SOA Reference Architecture" is one of the strength. It describes well the composing blocks of the EA and differentiates between conceptual, applicative and technical views. The domain "SOA Project" includes all issues including "SOA maturity model", "SOA Governance", "SOA Objectives & KPIs". For the domain "tool" the authors are giving capabilities that should be met for the different issues (e.g. SOA tools in design time and run-time, which is very useful in a concrete method). It is the only industrial method referring to and explaining high-level MDA. However, the whole method has a good coverage but stays relatively high-level without giving detailed explanation for most of the named issues.

### 3.2.9. Summary on SOA Methods Qualification

None of the existing SOA Methods is covering in full details all SOA domains. The qualification also says nothing about SOA Method quality, but only related to the coverage of SOA Domain Model. A root-cause for this consists in the provenance of these SOA Methods. For example, AVE for SOA from IDS has its roots in BPM and Modelling and not as such in components related to run-time or web-service domain. Or PIM4SOA is very much model & tool oriented and neglects completely the BPM domain and SOA project domain. The reason is clearly related to the root and objective of these methods which is based on specific viewpoints.

The methods having the largest span of mentioned sub-domains are ORACLE Unified Method for SOA (industrial) and SODD (academic). Most of the methods propose a top-down approach and underline the business benefit and the resulting high quality of SOA architecture as deliverable. Only the method of IBM and Mike Papazoglou justify the bottom-up approach in some circumstances. Deliverables in the INTEROP [Dou07] and ATHENA [ATHEN05] – projects also come to the result that a top-down method has more strengths than weaknesses.

Some methods are specially developed for the context of SOA (e.g. Zdun & Dustdar) others are derived from existing methods and adapted to SOA flavour (e.g. AVE for SOA).

MDA is sometimes mentioned, but never mapped to the different layers of abstraction. Also model types are sometimes mentioned, but not in a systematically way. Certain modelling languages and standards are more frequently cited or used than on others (e.g. EPC, UML, BPMN, BPEL and WSDL). If we compare how modelling and modelling notations are explained and used in the methods, major differences can be noted. The VbMF is using UML as modelling standard, whereas AVE uses EPC, SODD uses BPMN etc. Table 25 summarizes the evaluation of rated methods using the SOA Domain Model as framework:

**Table 25: Rating of SOA Methods according to the SOA Domain Mode**

| Legend: Almost Fully Covered: *** Largely Covered: ** Partially Covered: * Not Covered: No Star | | | | IDS Scheer | SAP | Zdun & Dustdar | Benguria & Larrucea et al | Papazoglou | IBM, Arsanjani | Oracle |
|---|---|---|---|---|---|---|---|---|---|---|
| Conception | Domains | SOA Sub-domains: Management of... | Nb | ARIS AVE for SOA | Enterprise SOA | Model-Driven Int. of Process | PIM4SOA | SODD | SOMA | Oracle Unified Method for SOA |
| SOA Method | Modelling | SOA Modeling Notation | 1.1. | *** | * | ** | ** | * | * | * |
| | | SOA Model Transformation | 1.2. | ** | | ** | *** | | | * |
| | | SOA Modeling Strategy | 1.3. | * | * | * | * | * | * | * |
| | Web Service | SOA Heartbeat | 2.1. | * | ** | * | * | ** | * | ** |
| | | SOA Security | 2.2. | * | ** | * | * | * | * | ** |
| | | SOA Decomposition | 2.3. | * | ** | ** | * | * | * | ** |
| | | SOA Patterns & Best Practice | 2.4. | * | ** | * | | * | * | ** |
| | | SOA QoS | 2.5. | * | ** | * | * | * | ** | ** |
| | | SOA SLA | 2.6. | * | * | * | | * | * | * |
| | BPM | Business BPM | 3.1. | *** | * | * | * | ** | * | * |
| | | BPM Knowledge | 3.2. | ** | * | | * | * | * | * |
| | | Technical BPM / BPM System | 3.3. | ** | ** | ** | * | * | * | * |
| | Tool | SOA Design Time | 4.1. | *** | * | * | ** | * | * | * |
| | | SOA Run-Time | 4.2. | * | ** | * | ** | * | * | *** |
| | | SOA Project Management & Change Mgt. | 4.3. | ** | ** | * | * | *** | ** | ** |
| | | SOA Process & Web-Service Simulation | 4.4. | * | * | | * | * | * | * |
| | Project | SOA Maturity Model | 5.1. | | | | | * | * | * |
| | | SOA Governance | 5.2. | | * | * | * | * | * | ** |
| | | SOA Objectives | 5.3. | * | * | * | | * | * | ** |
| | | SOA Phases | 5.4. | ** | * | * | | ** | ** | ** |

All presented methods have their right to exist with areas on which there are more complete or less complete than others. As we have these differences in coverage, the research need introduced in the first chapter is relevant.

For more detailed description of discussed SOA Methods, refer to Appendix B.

## 3.3. Evaluation of Research Artifacts by Data Collection Through Survey Design

### 3.3.1. Introduction to Survey

#### 3.3.1.1. Preparation of Survey

Online research methods are ways in which researchers can collect data via the internet. Many of these are related to existing research methods but re-invent and re-imagine them in the light of new technologies associated with the internet. With the increasing use of the internet, online questionnaires have become a popular way of collecting information. The design of an online questionnaire often has an effect on the quality of gathered data. There are many factors to take into account in designing an online questionnaire: guidelines, available question formats, administration, quality and ethical issues should be reviewed. Online questionnaires should be seen as a sub-set of a wider-range of online research methods.

There are several reasons motivating the decision to use the online questionnaire as preferred testing method. A few of the advantages and disadvantages of this method are [SRP02] [BNSSW+04][WKB89] summarized:

Advantages:

- The administrator has greater flexibility in displaying questions. Questions can be displayed with: Check boxes, Pull down menus, Pop-up menus, Help screens, Graphics.
- An online forum allows responses to be received more quickly from respondents.
- This method is cheaper to administer, as there are no costs associated with purchasing paper or other materials for printing. Postage costs are also mitigated.
- Since data is collected into a central database, the time for analysis is subsequently reduced.
- It is easier to correct errors on an online questionnaire, since the administrator does not have to reprint all the questionnaires for distribution.

Disadvantages:

- Not everyone has access to the Internet, so the response rate may be limited;
- Many people are not receptive to completing questionnaires online [PRCLM+04].
- Studies indicate that the population that responds to online questionnaire invitations are generally biased to younger people (demographic representativeness issue) [PRCLM+04].
- Response rates are frequently quite low and there is a danger that they will continue to drop due to over-surveying of web-users.

Three main factors namely respondent ability, respondent motivation and questionnaire design determine the success of the questionnaire and the likelihood of achieving decent levels of response [GFFCM+04].

### 3.3.1.2. Research Questions

Based on these considerations and on the literature about setting up questionnaires, a web questionnaire has been chosen for the survey, mainly because of its efficiency (quick collection of responses, low effort for analysis and low cost). The objective of the qualitative survey is three-fold: first the motivation for the present research should clearly show the need also in industrial practice, second, the state of the art-analysis should be completed and refined by opinions based on practical experience. Third, the identified issues with the related questions should be given to practitioners for feedback.

In order to cope with research question 1 about SOA Methods identification and characterization, the whole SOA domain model with its issues has been transformed into questions to get insights in industrial expertise and possible solutions. Second, research question 4 about suited candidate modelling languages for SOA implementation is investigated.

Beside the enumerated SOA Domain Model issues, a more introducing generic part was asking for known SOA Methods, the complexity to use and apply them and about the popularity and awareness of academic SOA implementation methods.

In relation with the posed research question 4 about suited candidate modelling languages on different levels of abstraction, the level of knowledge including application & satisfaction towards available SOA Methods has been asked. Here, knowledge gathering objectives as well as testing objectives are set:

A.) Knowledge gathering objectives can be deducted into three sub-questions

> A1) which modelling notations seem the most suitable for SOA implementation?
> A2) what are critical success factors? Is BPM knowledge in particular a critical success factor?
> A3) what is the degree of popularity and awareness of academic SOA implementation method proposals?

B.) Testing Objectives

> B1) is the proposed SOA domain model complete?
> B2) is the lack of method perceived as an issue? Is the subject of SOA method perceived as complex and do users know where to start?

The complete questionnaire template is available in Appendix A.

Other research questionnaires about SOA implementation such as from Viering and Legner [VL09] conclude that SOA implementation is still on-going and relevant on a broad level.

### 3.3.1.3. Data Collection

To test the questionnaire on content, design and relevance, a trial group of three specialists being subject matter experts filled the questionnaire and provided feedback.

The empirical study was performed from August 2008 to January 2009. The survey was accessible by following a web-link, available 24h/7 and recording all entries in a database. The chosen channels for the announcement of the survey were professional communities related to SOA including qualified profiles of managing IT members (i.e. BPtrends [BPTRE09], IT Nation [ITNAT09] and SOA Know-How [SOAKN09]). Due to this specific target, the issues of demographic relevance and of respondent's ability can also be strongly reduced. CIOs as owner of the overall IT strategy, of which SOA method can be considered a sub-part, should be able to respond in a competent manner. By nature, CIOs are also used to novel research technology. The motivation issue has been faced by the promise to provide an executive summary report of the survey results by e-mail to respondents. It is true that the questionnaire was more complex and longer to answer than other simple industrial questionnaires, but this could not be avoided due to its academic background and objectives.

### 3.3.1.4. Limitations of conducted survey

With the chosen approach, it was unfortunately not possible to calculate a ratio of participation. Furthermore, it was not possible to measure how many visitors in that timeframe have seen the link, clicked on it and then decided to participate.

A first attempt getting access to worldwide IT specialists in companies was to ask IT market research providers to participate e.g. Gartner, Forrester, AMR Research. Unfortunately, this initiative failed as the questionnaire was rated too academic and too time consuming to fill. Out of the total number of answers (79) 54 relevant ones were selected by eliminating responses not being serious or complete (less than 80% of answered questions). The total population of the sample is 54. However, in the next sections, figures might sometimes not match 100% as respondents within the 54 might not always have responded 100% of all questions. The population sample will be shown in each statistic with n=x. The top five countries to respond were Luxembourg (16,7%), USA (16,7%), Germany (14,8%), Belgium (11,1%), Australia and Brazil (9,2%). The respondents' countries are obviously correlated with the distribution over countries of the members of the community of the three BPM/SOA websites.

72,2% of respondents are Managers, Directors, CIO/CPOs or CEOs. The profiles show clearly that those who responded have a good overview of the subject. Obviously most of the respondents are also profiles who will decide about implementing SOA and how this will be done. This is on the one hand a strength and positive aspect because the survey collected the viewpoint of deciders, but on the other hand this might represent also a weakness as SOA analysts and programmers are underweighted. On the other hand, the responsible managers have filled the technical questions together with their analysts and architects. Unfortunately, the research design was not able to provide a validation that respondents have well understood the questions and eventually have referred to analysts being more competent to answer the questions. However to reduce this risk, it was proposed in the survey introduction to ask questions by email to get clarification if necessary. Some CIOs in Luxembourg have asked to fill the questionnaire with my assistance to ensure a correct understanding of questions.

In total, the number of 54 respondents is not sufficient to deduct highly statistically significant final conclusions for a quantitative survey with the final objective to validate issues. Furthermore, the sample of respondents can be considered as interested and experienced in BPM and SOA. Those, who have no interest or belief in SOA, also had no interest in responding to the questionnaire as this was a time consuming commitment.

### 3.3.1.5. Critical discussion on questionnaire design

The proposed steps by Walonick [Wal04] for a research survey have all been followed such as 1.) Design Methodology, 2.) Determine Feasibility 3.) Develop Instruments, 4.) Select Sample, 5.) Conduct Pilot Test, 6.) Revise Instruments, 7.) Conduct Research, 8.) Analyze Data, 9.) Prepare Report.

The design of the questionnaire has been done with the objective to reach interested target groups from specific websites. The whole questionnaire has been structured following the domains of the SOA domain model. Due to the large number of issues to get information about, it was not possible to shorten below 36 questions. The goal of the questionnaire has

been communicated and an incentive to fill the questionnaire has been given. On the other side, the filling of name, position and e-mail address was also asking for a limited personal amount of information – which could have threatened some respondents. Unfortunately, no survey design experts for the survey design were available for detailed review. However, the following principles of questionnaire design have been applied:

- Non-threatening questions,
- Multiple responses possible,
- using 4-likert scales to avoid "neutral" responses,
- Reduction of ambiguity in the question understanding to a minimum by short questions and well-known and industry standard terminology (IEEE Definitions),
- Variability in responses for statistical analysis,
- Grouping of questions in related domains,
- Mistake reduction on pre-assumptions such as "knowledge" pitfalls by adding always a free-text option "other",
- Question wording as objective as possible, not implying a desired answer.

In case of misunderstanding of wording in questions or non-comprehension, it was always possible to ask a question for clarification by e-mail. Standard Terminology has been used to prevent misinterpretation. Where possible, the free-text option "other" was included into the response options. For the objectivity of wording, the enumerations of methods or modelling notation were always in alphabetical order. However, the question 36 is formulated a bit into the direction that respondents could tend to answer more into a certain direction "yes, I agree" as psychologically it is easier to say "yes" than "no". On the other hand, again the "other" option has been applied and valuable responses were given to challenge the SOA Domain Model and also enhance it with issues that were not explicitly addressed.

## 3.3.2. Questionnaire Results

### 3.3.2.1. Respondents profile

As said in the introduction to this chapter, the profiles are rather senior: 72,1% of respondents are Managers, Directors, CIO/CPOs or CEOs. The profiles show clearly that those who responded have a good overview of the subject. Obviously most of the respondents are also profiles who will decide about implementing SOA and how this will be done. The survey provides the perspective of individuals from a wide range of industries as shown in the figure below. 74% of answers come from headquarters and 26% from subsidiaries.

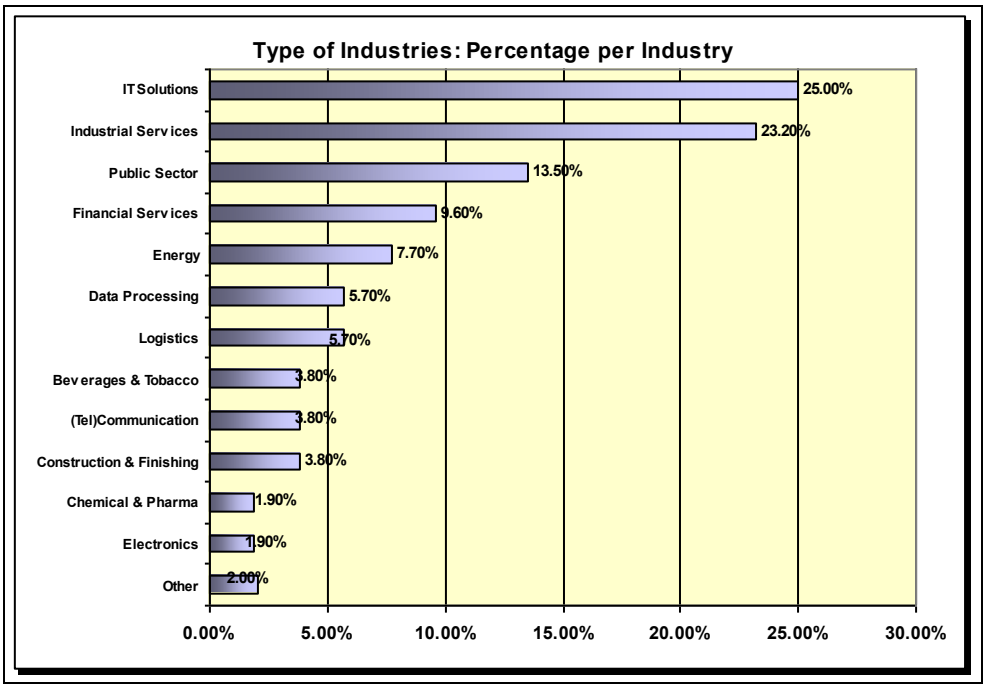



**Figure 30: Type of Industries: Percentage per Industry (n=52)**

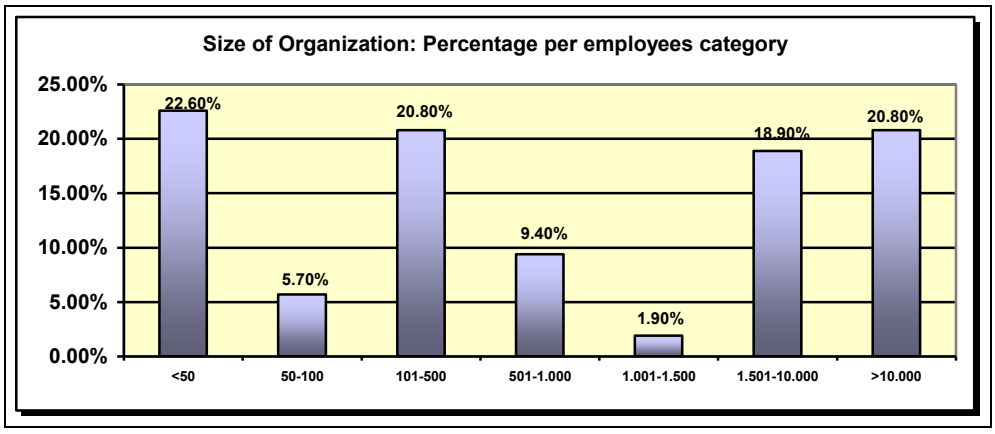



**Figure 31: Size of Organization: Percentage per employees' category (n=53)**

An important criterion for the utility of SOA implementation is the size of the company/organization. With the size, usually also the number of applications is increasing. The panel of respondents have the following size and number of applications:

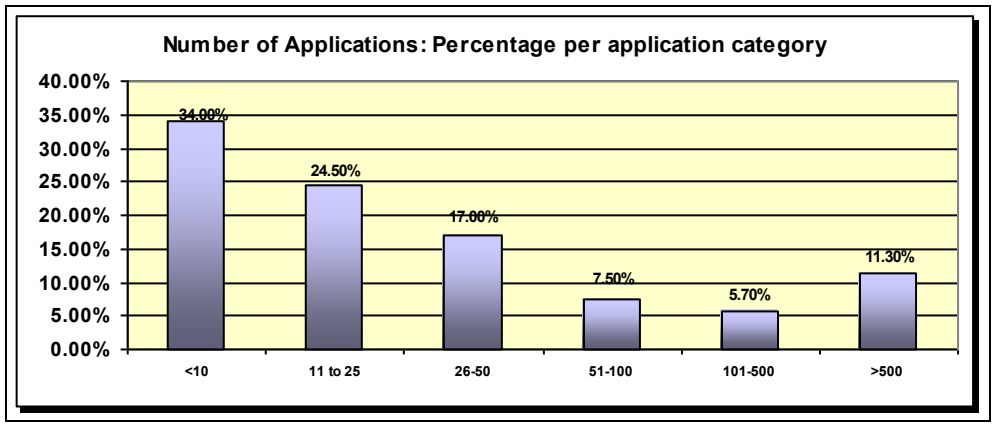



**Figure 32: Number of Applications: Percentage per application category (n=53)**

#### 3.3.2.2. The Respondents Context for SOA

In this section, we analyse the global situation and context (maturity) of respondents regarding knowledge and use of SOA.

98 % of the participants know the SOA concept, whereas nearly 50% started to know about SOA within 2005 and 2006.

96,2% of respondents will use the SOA concept against 3,8% who decided not to use the SOA paradigm in the IT Strategy. This ratio shows clearly the relevance of thinking more in detail about possible ways of usage in the organizations. As mentioned in the beginning, we can assume that only BPM and SOA aware respondents were interested in contributing to this research.

Out of the 96,2% of respondents deciding to go for SOA, 50% have planned to go for the project 26% are involved in an on-going SOA project, 10% have already finished the project and 14% are in the discovery phase of SOA (investigating what it is and how it can be tackled in the best way). In summary, 64% have not started whereas 34% have started or finished.

If we examine the 10% of respondents with already implemented SOA project, these are very big companies with a clear business case and a high level of education and maturity around SOA technology and Business Process Management.

When asking about the benefits that SOA can bring to organizations and companies, the usual benefits are nominated. An interesting result is the ranking starting with the strongest argument for the implementation of SOA:

1. Flexibility and Agility in IT Architecture and the possibility to re-use services
2. Business and IT alignment by common views and language
3. Reduction of IT cost
4. Enforcement of a “process” thinking
5. Re-utilization of Business Process content
6. Enforcement of data quality

Opposed to benefits of SOA are also challenges faced. Here are the reported challenges in decreasing order of importance:

1. ROI difficult to calculate
2. Subject is complex
3. Missing approach and where to start
4. Tangible benefits hard to identify
5. Knowledge & right profiles
6. Organizational alignment
7. Change Management
8. Top-Management Buy-In
9. SOA Governance

Interestingly the respondents were much more aligned on what are the biggest advantages than on the challenges. Within the list of challenges it clearly states the issue on missing

approach of the complex subject. The proposed artefacts as research contribution will help to solve this problem.

## 3.3.3. Detailed Results on testing the SOA Domain Model

### 3.3.3.1. SOA Domain: Modeling

According to our state-of-the-art analysis, EA is an entry point and is playing and important role in the context of SOA implementation. The thought about how an EA can support the IT strategy is a key success factor to include also the SOA concept in the IT strategy. Finally, EA is key for SOA implementation, as method, modelling, process management, abstraction layers views and linked components need to be considered. The list of EA presented in the questionnaire was populated with the most common in academia and industry. It is highly interesting to know which standards are known or used by industry. If an EA is used in practice, it is also interesting to see if the respondents are satisfied with it. Therefore, for the modelling domain, most of the questions asked for one answer among the following possible ones: not known, known, used meeting expectations used not meeting expectations. The result clearly shows that some EA (e.g. CEN ENV 400003, GRAAL, GERAM, TOVE, TEAF, AKM) are not known and therefore not used at all. On the other hand there are clearly EAs that are known and used by most of the respondents (e.g. Zachman, ARIS, 4+1 View Model of Architecture, MDA, RUP).

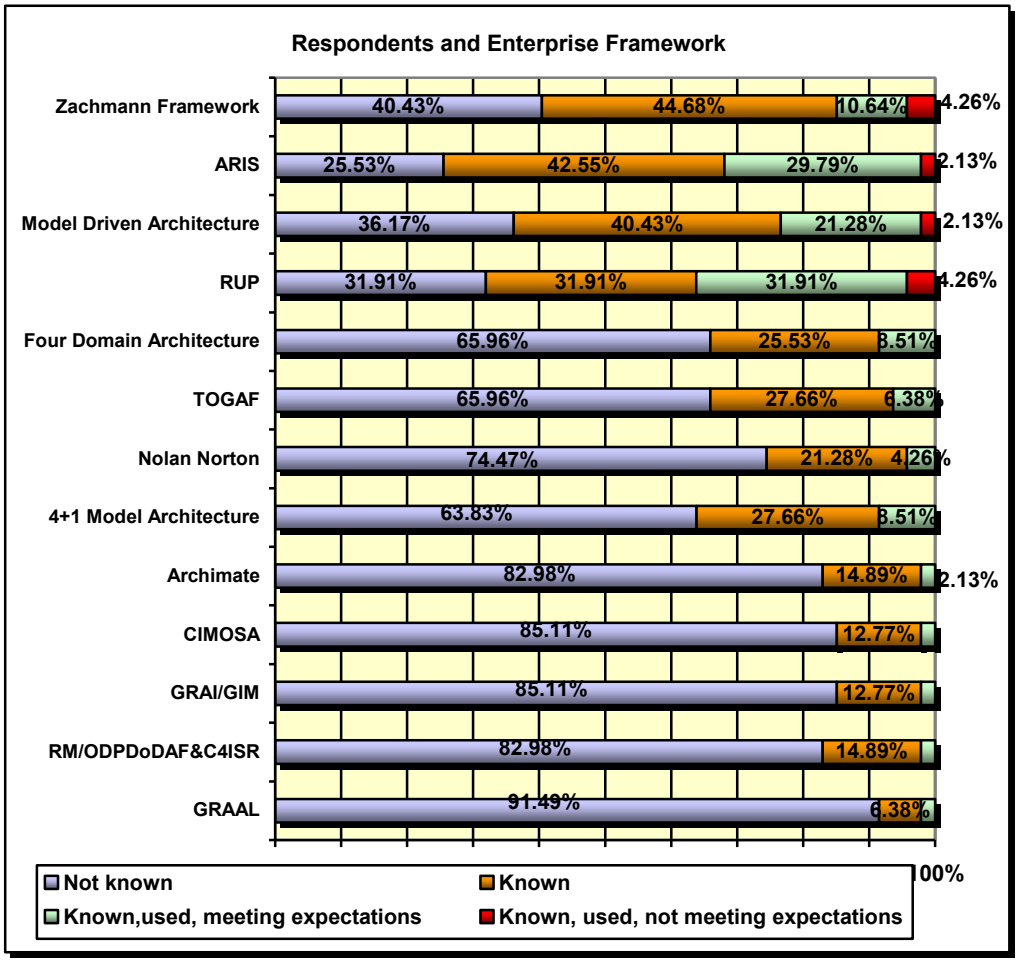



**Figure 33: Respondents and enterprise Framework (n=54)**

Notably, there is a relationship between the country of origin of the companies and the known and used EA. Respondents from German speaking countries (e.g. Austria, Germany, Luxembourg) have a clear focus on ARIS, whereas US related respondents are more in favour of Zachman, MDA or RUP, which are standards that have been defined and are maintained in the United States. Some EA have also regional or country related roots e.g. CIMOSA in France or ArchiMate in the Netherlands and therefore, a limitation of our survey is the under-weighted proportion of French and Dutch respondents.

Similar to EAs, modelling languages are important to analyse in the context of a SOA implementation. Which are the modelling languages suited to accompany conceptual processes with the objective of SOA implementation? As we take a processes-oriented viewpoint, candidate notations or modelling languages elaborated in chapter 2 were asked for practitioners' feed-backs.

In general, strategic model types such as e$^3$value, Balanced Scorecard (BSC) or VACD are less known and used than business process requirement languages such as Business Process Modelling Notation (BPMN), Event driven Process Chain (EPC) and UML Activity Diagram or than technical process implementation languages (such as BPML or WSDL).

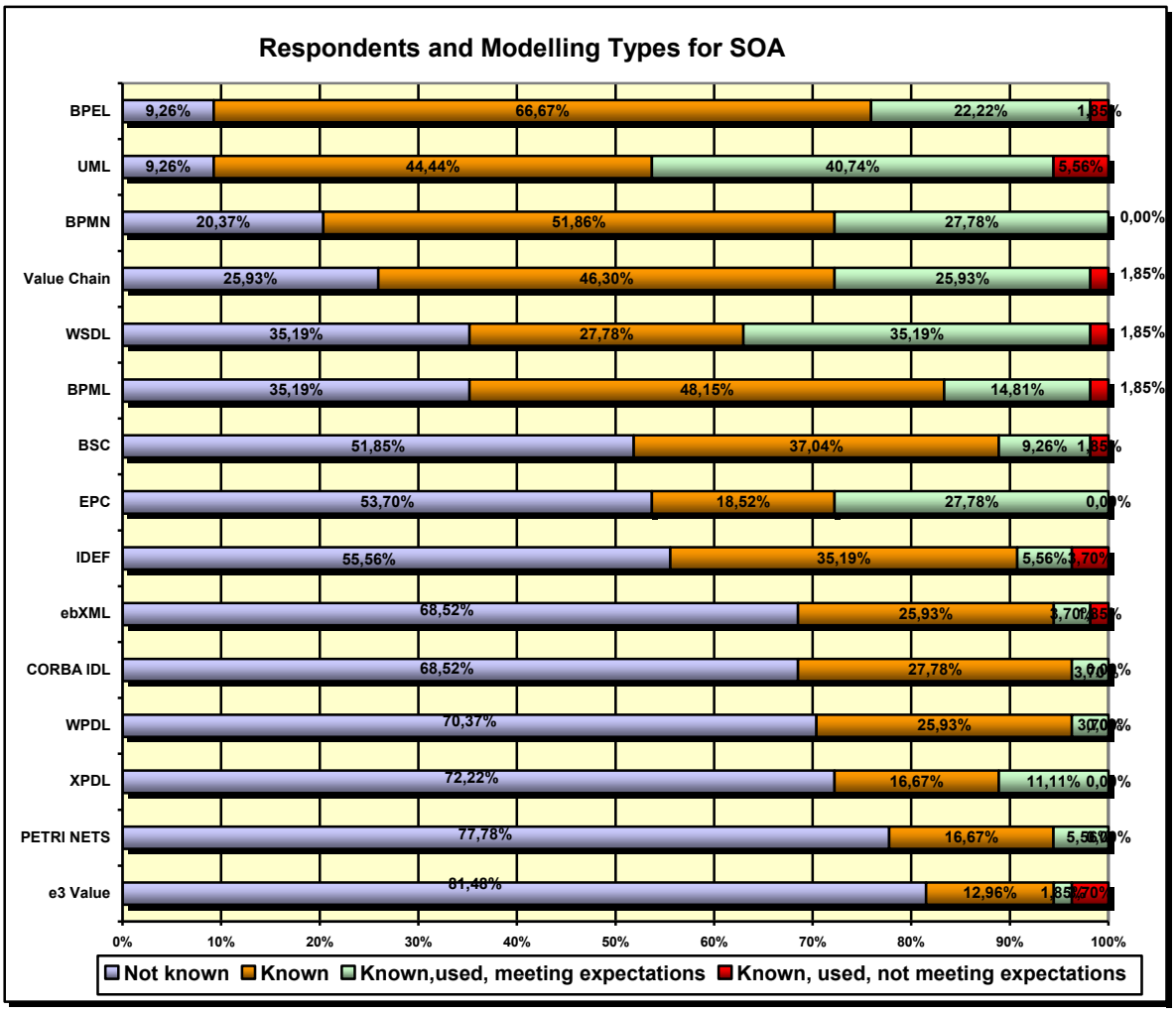



**Figure 34: Respondents and Modelling Types for SOA (n=54)**

Some modelling languages are not known and used at all[1] : Archimate, BOP, EEML, EKS, Grai/Gim, IEM/Mo2Go, JPDL, Memo, Metis, Meml, Pim4SOA, PIF, PSL Core, SADT, SPEM, Testbed, UEML and Yawl.

Clear trends are visible about modelling languages usage on the three different levels of abstraction (Strategy, Processes, IT). For Strategy, the most known and used model type is the BSC model and Value Added Chain model. Most of business requirements at the process level are captured through BPMN, BPML, EPC, IDEF, UML Activity Diagram. For IT or implementation languages, BPEL, WSDL, WPDL are particularly often known and used. Regarding the way SOA is implemented 57,4% of respondents have chosen the top-down approach, 20,3% meet-in-the-middle and 22,3% Bottom-up. The result shows a clear trend towards top-down approach and even more decide for meet-in-the-middle than for bottom-up.

The MDA approach of the OMG for software development is gaining popularity. The way abstraction levels are defined and what types of models are used is also important for the context of SOA implementation. Most of respondents know MDA for software development (46,30%) and also use it with satisfaction (16,67%), not meeting expectations (3,7%) and a bit more than a third (33,33%) do not know about MDA. In the context for SOA developments the knowledge about MDA is similar and approximately 13% claim also to use it in this specific context. Notably, MDA is known and used successfully by the respondents coming from the leading worldwide IT service providers. They have a high level of knowledge and maturity in software development and also apply MDA to their SOA implementation approach.

A principle of MDA is the automatic transformation of technical models (such as UML models) into code. The automation paradigm is also advocated in the context of SOA. The question has been answered by 35,0% of respondents, but most of them rated the question as not applicable. Again here, nearly all of the respondents are in the IT industry. Still, on a very low level, one can recognize which translations are used more often than others. For SOA, more automation is reached the closer one comes to the detailed level of PIM and PSM (related to MDA method). Out of the small population answering to this question, respondents have successfully used BPMN2BPEL (25,00%), BPEL2WSDL (20,00%), UML2WSDL (10,00%), EPC2BPEL (10,00%), EPC2BPMN (10,00%), UML2BPEL (10,00%), EPC2UML (5,00%).

#### 3.3.3.2. SOA Domain: BPM

Another important dimension, the BPM, is considered as critical success factor and enabler. Therefore, in total 83,3% of respondents manage (completely: 46,3% or partly: 37,0%) their processes in a real BPM programme including strategy, design, implementation & controlling.

Within their BPM initiative, various usage scenarios are covered or addressed:

1 Meaning that more than 85% of respondents do not know nor use it.

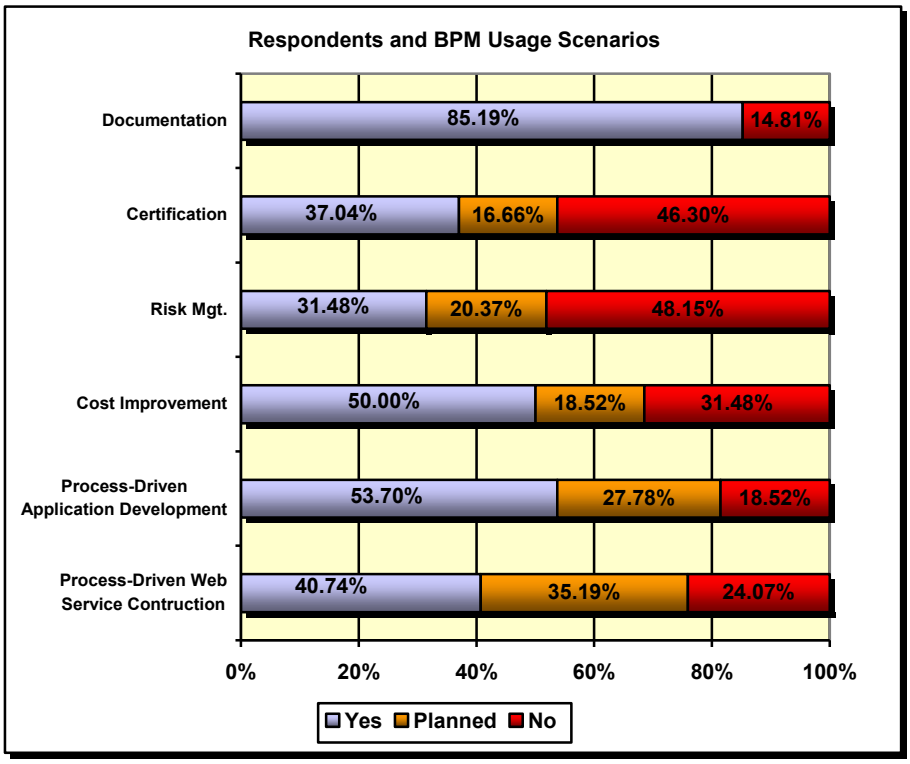



**Figure 35: Respondents and BPM usage Scenarios (n=54)**

Most of respondents have already documented processes (85,19%) and use BPM also for other objectives e.g. certification (37,04%), risk management (31,48%), cost control (50%), process driven application management (53,70%) and process-driven web-service construction (40,74%). In the context of SOA, it is very interesting to observe the planned scenario for the two last cited with 27,78% and 35,19%. Consequently, more than 77% of respondents are using or have planned to use processes for the web-service identification and construction. Furthermore, the planned process-driven web service construction of 35,19% is the highest value for the planned usage scenarios in BPM. This is clearly the area with the biggest increasing potential of re-utilisation of BPM content.

Generally, the BPM knowledge is rated as very important with 90,74% for SOA implementation. Only 9,26% of the respondents rate it neutral (7,41%) or as not important (1,85%).

#### 3.3.3.3. SOA Domain: SOA Project

Maturity models can help to identify the current status and can support thoughts on targeted maturity and the way to get there. Originally developed by CMMI, maturity models are these days also proposed for SOA maturity. Only 20,4% of respondents use a maturity model for SOA. Exactly half of these respondents (10,2%) declare to use the Gartner SOA Maturity model, the other half (10,2%) is using their own developed model.

The Return Of Investment (ROI) is a key figure for IT projects decision making. The biggest challenge as indicated by the respondents is also substantiated in the following result: 77,78% of respondents did not succeed in calculating the ROI. The ROI calculation is related to the business case the companies/organizations have for SOA: 46,30% argue to have a strong business case for SOA with 51,85% claiming to possess the right skills to understand SOA

and 44,44% with the right skills to implement SOA. 48,15% of respondents need external consultants to implement SOA.

An important issue to address is IT project management that could be adapted to manage the SOA project. 72,2 % use their own project management methodology, 18,5% follow PMI and 9,3% follow Prince2. Within the 72,2%, a considerable number of respondents has adapted and mixed PMI and Prince2 for their specific needs.

Next, the respondents were asked to evaluate a list of SOA methods that resulted from our state-of-the-art analysis of all current availably SOA methods in the academic and practice worlds, as shown below:

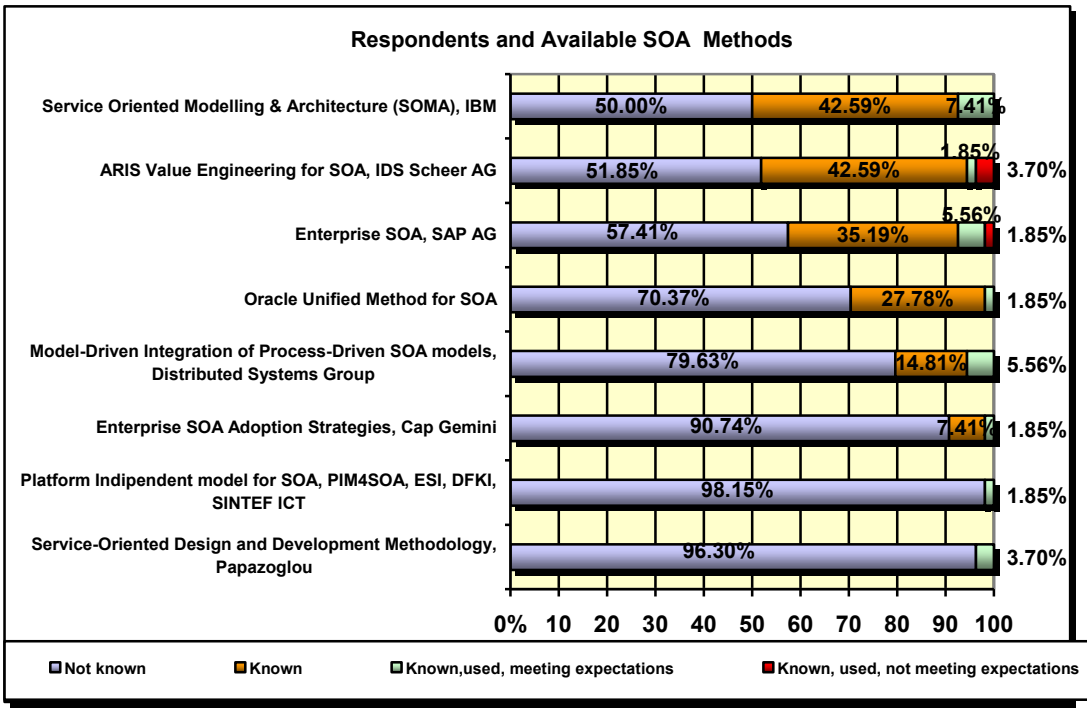



**Figure 36: Respondents and Available SOA Methods (n=54)**

In general, most of respondents are not aware of the wide range of existing methods. The most known methods are industrial ones e.g. IBM (known by 42,59%), IDS Scheer (42,59%), SAP (35,19%) and ORACLE (27,78%). The academic proposals are even less known than the industrial SOA methods. Unfortunately, the number of reported successful application of such methods is too low to deduct reliable findings. IBM was the first IT company to invest in SOA run-time engines and SOA method (SOMA). Therefore, their solutions and methods are more known than these of the competitors. (The business motivation model has not been introduced in the questionnaire as it has been considered too new and not mature enough.)

The root cause for the weak knowledge on SOA methods is related to the fact that 87,04 % of respondents rate SOA method as a very complex issue and not easy to tackle at all. If IT-service providers are taken out of the panel, the figure is increasing up to 98,15%. As already mentioned, the IT service-providers have a good understanding of mainly technical SOA knowledge and therefore see in most of the cases no huge complexities to solve.

An important aspect to accomplish successfully SOA projects is related to identification of specific SOA objectives, Key Performance Indicators (KPIs), SOA drivers and Critical

Success Factors. Only if this strategic part is well understood and formalized, SOA can become a real success story. Without clear objectives and ways to measure it, the business case will be weak and the calculation of ROI very difficult. Within the respondents, 16,67% have this formalized strategic SOA dimension, 37,0% have it partly and 9,26% have planned to establish it. 35,19% have no such written objectives.

### 3.3.3.4. SOA Domain: BPM Design Time Tools & BPM Runtime Tools

BPM is a key enabler for SOA. Therefore processes need to be supported by robust tools. This is true for the so called "design time" environment as well as for the "runtime environment" What tools or platforms are known and successfully used on both levels? The following chart gives an overview of the respondent's situation:

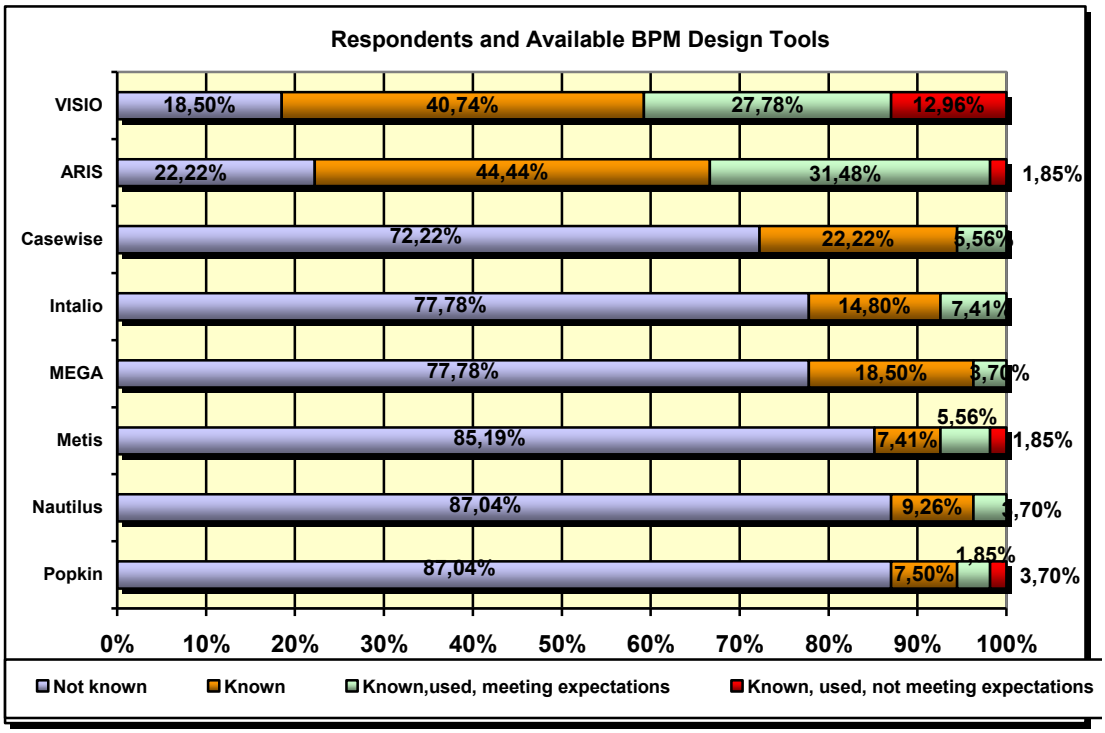



**Figure 37: Respondents and available BPM Design Tools (n=54)**

On the "design time", it clearly shows ARIS platform ahead as well as known and also used successfully. Furthermore, Visio is also well known and used, but with a higher rate of non-satisfaction related to BPM and SOA modelling. Visio is still more considered as drawing tool that can be used and mastered very quickly than a real BPM design tool.

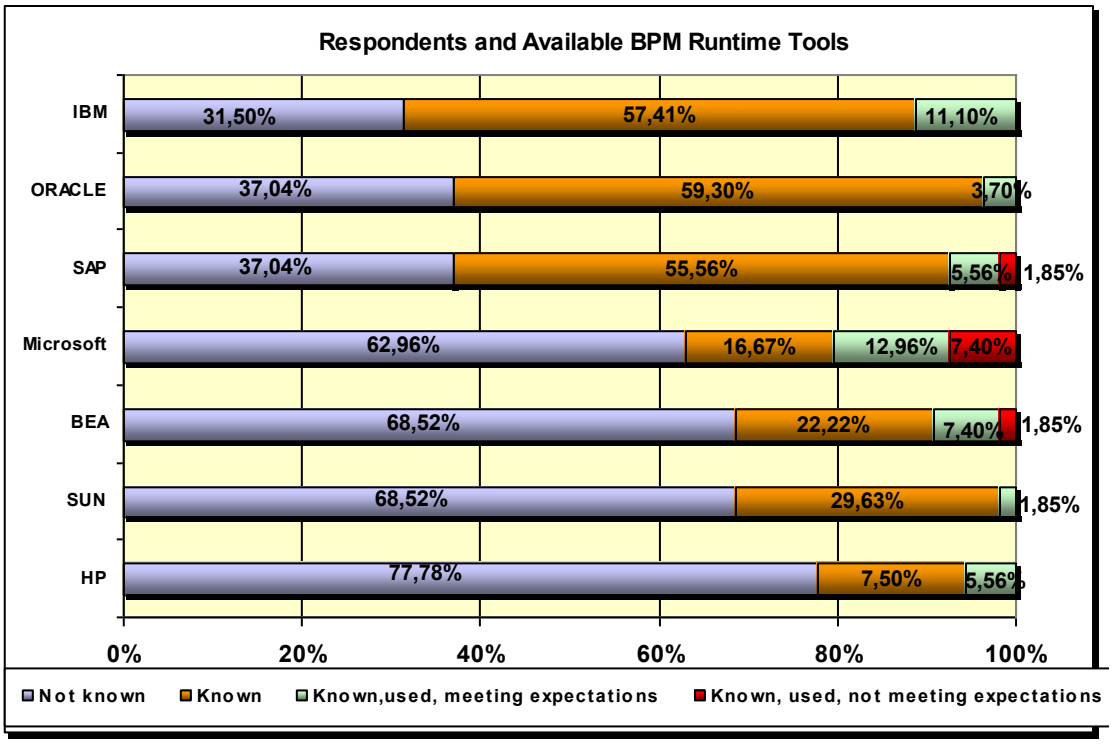



**Figure 38: Respondents and available BPM Runtime Tools (n=54)**

On the "runtime environment", IBM, Oracle and SAP are most known and used for implementing and running BPM. The BEA products as well as SUN were taken over by Oracle, which consolidates a bit the runtime environment providers. Within other runtime environments, e.g. Mircosoft or HP is cited.

### 3.3.3.5. SOA Domain: Web Service

A central domain in the SOA paradigm is for sure the service concept. Related to services, 63,16% of respondents answered that service orientation is part of their business strategy. This is partly true for 21,05 and 15,79% argue their business strategy is not service oriented.

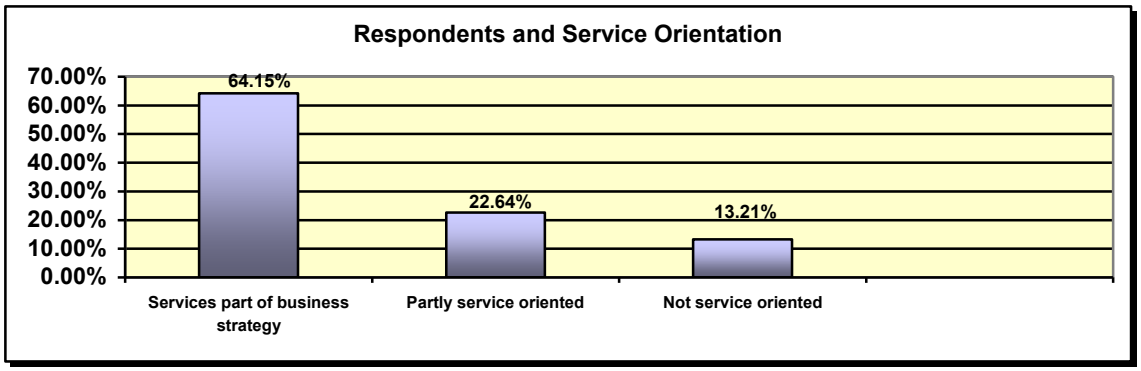



**Figure 39: Respondents and Service Orientation (n=53)**

Interesting in that context is the IT situation of respondents: 34,21% are in full outsourced mode, 5,26% partly and 60,53 have their IT in-house. If we analyse the other way around, more than a third of respondents (answers: "yes" 26,32% and "partly" 10,53) deploy business web services measured by a Service Level Agreement (SLA) to other companies.

81,5% of respondents use already web service technology, 18,6% don't. Web-service technology can well be used just to interface applications. It is not an indicator for service oriented architectures.

A frequent discussed question in this context of web-service development is the approach of web-service construction: Is the business asking for new services (top-down) or is the IT developing services to present these to the business (bottom-up)? The respondents agree with 77,78% that business is asking for new web-services (top-down) to better support their business processes.

Web service security is also considered as an important issue to tackle. Within SOA security management authentication, authorization and identity management need to be addressed. The following graphic illustrates the results:

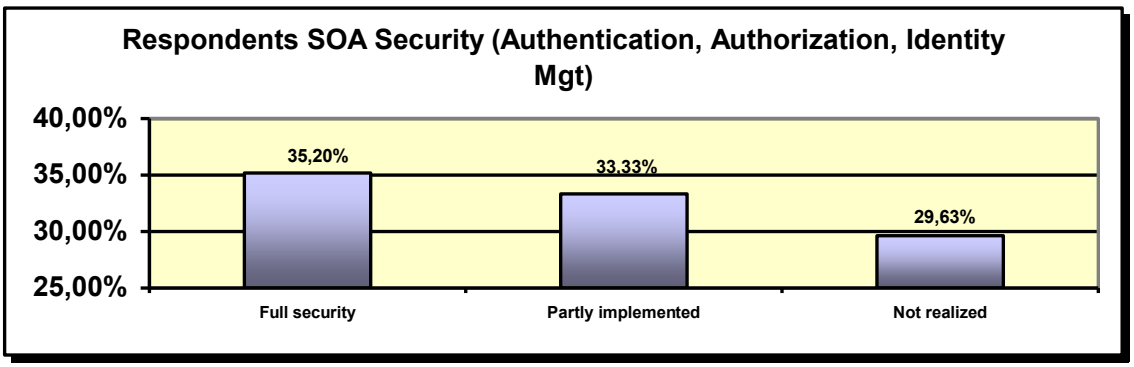



**Figure 40: Respondents SOA Security (n=54)**

All respondents having answered "no" have so far not started their SOA project.

Web service granularity and decomposition is still for 87,04% a major issue. Only 12,96% think this is no issue for them. Again, 100% of these respondents arguing granularity is not an issue are within the group of IT service providers having already implemented SOA.

Data itself is an important part of SOA management and implementation. Therefore it is useful to master and control data appropriately. The following results were gathered about Data Management:

37,04% have a data management programme implemented, whereas 31,48% have it partly. (No data management for 31,48%)

48,15% of respondents claim to master the interfaces between applications, whereas 40,74% do partly. (No interface mastering for 11,11%)

Only 37,04% have automated application interfaces, 48,15% have it partly. (No interface automation for 14,81%)

#### 3.3.3.6. SOA Domain Model

Finally the outcome of the state-of-the-art analysis needed to be tested on completeness related to industrial experience.

90,74% of respondents agree that the presented SOA Domain Model is reflecting all domains to consider for an exhaustive SOA implementation method based on a process-oriented approach. Within the 9,26% not agreeing, respondents were pointing to change management or top management support. The mentioned issues are part of the SOA project management domain and are addressed in the model. Some other respondents were pointing to related approaches e.g. Web-Oriented Architecture (WOA) or Representational State Transfer (REST) approach.

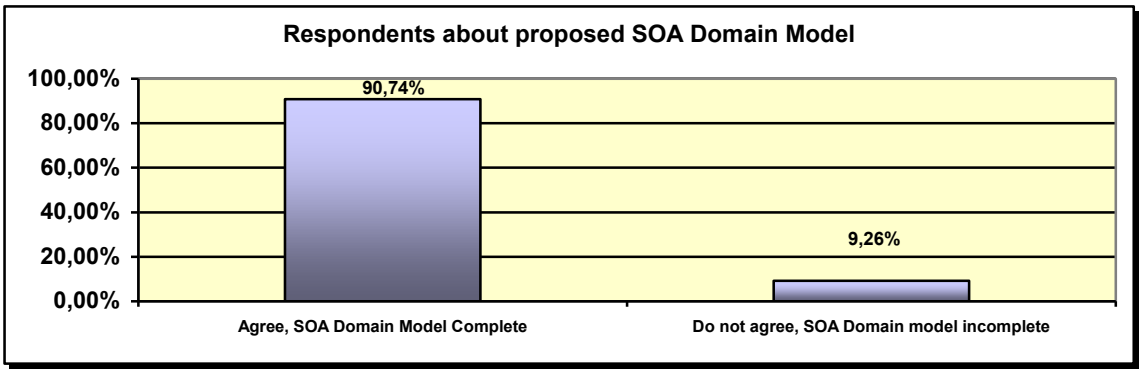



**Figure 41: Respondents about proposed SOA Domain Model (n=54)**

Smith [Smi08] is arguing that "WOA, like SOA, is an architectural approach to system design, though WOA is resource-oriented rather than service-oriented. While the core SOA design unit is a reusable service that fulfils a distinct business function, resource-oriented services are more limited and data-focused. SOA and WOA work at different layers of abstraction. SOA is a system-level architectural style that tries to implement new business capabilities so that they can be consumed by many applications. WOA is an interface-level architectural style that focuses on the means by which these service capabilities are exposed to consumers. Governance, quality of service, security, and management are of equal importance, whether the functionality is being delivered via SOA or WOA."

Therefore, WOA and REST are approaches standing for their own. They could certainly add value for specific questions.

### 3.3.4. Summary on Questionnaire Results

In this chapter, we presented the results of a survey on the knowledge and practice of SOA in industry. 54 respondents gave complete and relevant answers. The answers are satisfyingly representative of companies from around the world. From the results obtained, we can draw some general conclusions.

Regarding the statistical significance of the respondents, a more world-wide participation would have been valuable. Unfortunately, the objective of benchmarking the results between

different industries has not been achieved because the total respondents' number per industry was in total too low to get statistical significant results.

Several reasons have impacted the number of respondents: First, due to question deepness, filling the questionnaire required substantial effort and time. Second, respondents needed a certain level of knowledge, maturity and understanding of the topic to contribute seriously to the survey. Third, the financial crisis 2009 stroked just in the period of launching and advertising the questionnaire and induced, as we could observe in our contacts with the sector, a swap of priorities from strategic IT investments (among which "SOA implementation" projects) towards a more "IT cost control" focus.

Overall, the results show clear tendencies and underline statements from the state-of-the-art analysis and will lead into two detailed field trial studies to apply the SOA domain model for further refinement. Related to the research question number three posed at the beginning of this chapter, we can conclude the following:

A.) Knowledge Gathering Objectives

A1) Which modelling notations seem the most suitable for SOA implementation?

> The modelling notation candidates mentioned as the most appropriate are BPEL, UML, BPMN, Value Chain and WSDL. This result is matching with the state-of-the-art research in chapter 2. Other notation usage depends on countries or regions such as EPC modelling is well known for German speaking countries using mostly IDS Scheers' Tool ARIS.

A2) Are the principles "model-driven" and "process-oriented" considered as important?

> 90,74% of respondents rate BPM as critical for SOA Implementation. A clear trend shows which process model notations are successfully used for SOA implementations. Process knowledge will in the future be re-used by 35,19 % to do process-oriented web-service construction.

A3) What is the degree of popularity and awareness of academic SOA implementation method proposals?

> Academic SOA Implementation Methods are de facto unknown in industry and unfortunately also not used. IBM as first industrial service provider on the market for SOA solutions, their method is the most known and used.

B.) Testing Objectives

B1) is the proposed SOA domain model complete?

> Regarding the validation and completeness of our preliminary SOA domain model, 90,74% of respondents agree that the presented model is reflecting all domains to consider for an exhaustive SOA implementation method based on a process-oriented approach. Within the 9,26% not agreeing, respondents were pointing to change management or top management support as lacks. However, the mentioned issues are

already addressed in our model as a part of the SOA project management domain and are more generic nature and applicable to any other project too.

WOA and REST as described above were mentioned as missing topics but it can be considered that these approaches are standing for their own.

B2) Is the lack of method perceived as an issue? Is the subject of SOA method perceived as complex and do users know where to start?

Two out of three top issues related to SOA are “complexity of subject” and “missing method and where to start”. 87,04% rated SOA implementation method as complex. If IT providers are eliminated out of the panel, the percentage is increasing to 98,15%.

Finally, we can conclude that respondents do not rate available SOA method proposals as insufficient, which is clearly confirming the need for a SOA engineering method. Moreover, the SOA Domain Model seems to be “de-facto complete” and candidate notations for a process-oriented approach are clearly identical to the state-of-the-art research findings.

# CHAPTER 4

# RESEARCH CONTRIBUTION:

# A SITUATED SOA METHOD ENGINEERING FRAMEWORK



Chapter 4 is the second chapter on research contribution artifacts. First, the configuration process for SOA situational method (section 4.1.) is created, formalized and explained in detail.

Finally, the last outcome is the formalization of fragments (section 4.2.). The SOA Domain Model and ME principles are applied for formalizing an available standard SOA Method piece with the objective to demonstrate the formalization into a method fragment. This has been shown on the selection of process models for different levels of abstractions but only for the SOA domain of modelling. A summary at the end of each section concludes the most important findings.

## *4.1. Artifact 3: Configuration Process for SOA Situational Method*

First, an alignment model is presented to explain the relationship between the SOA Domain Model and ME. Based on this meta model, parts of 2 available SOA implementation methods are formalized into method fragments. We will concentrate on the model-driven and process-oriented part. It is explained, how this is done. Additionally, supporting tools and guidelines are presented to facilitate the application into concrete cases. These concrete cases are detailed in chapter 6.

### 4.1.1. Relationship between SOA Domain Model and Method Engineering

The SOA Domain Model introduced earlier is summarizing criteria's identified in the state-of-the art with the objective of implementing a process-oriented SOA. The criteria's related to the SOA domain model have been defined and described in the context of SOA. Therefore, a link between SOA domains with its sub-domains towards method fragments needs to be done.

Only this way, it is possible to check what coverage of criteria's has been achieved in the application of the situational method in a specific project application. The following model cannot be generalized for all SOA Domains, as only the SOA Domain "Modelling" has been formalized in fragments and applied in the action cases. Further work (section 7.3.) could investigate further into the formalization of method fragments in other domains.

The Class "SOA Domain Model" includes sub-domains. These sub-domains have been defined and described earlier. The alignment model below integrates attributes to unambiguously describe and classify the SOA sub-domains. Every sub-domain is related to an "activity" (refer to section 2.6.) which is a term to "neutralize" the semantics of vendor specific method fragments. Such an activity can include one or more available method fragments. One specific method fragment includes a process fragment and a product fragment. A product fragment is input/output to one or more process fragments. Figure 42 shows the link between the SOA Domain Model and ME terminology:

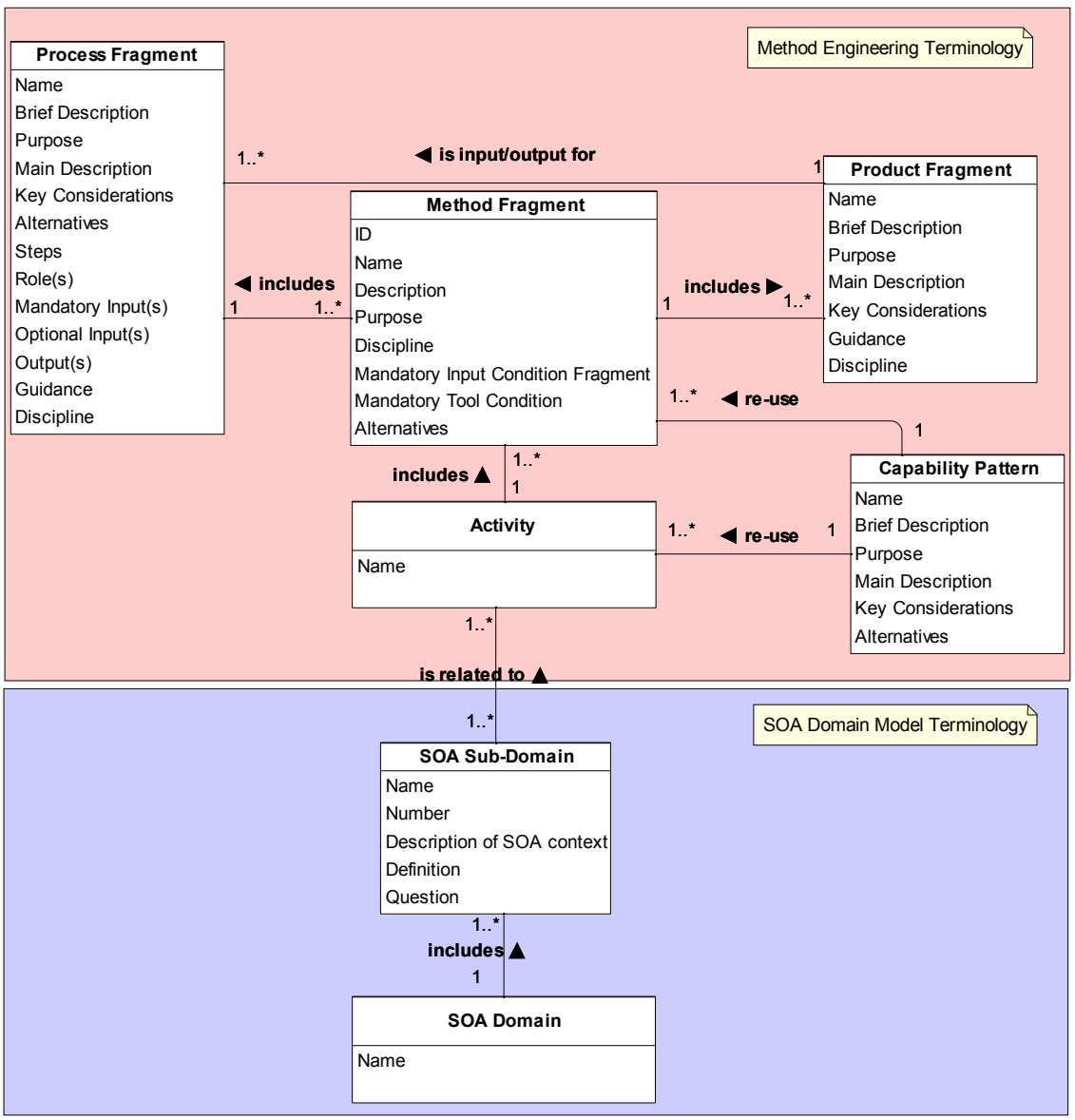



**Figure 42: Alignment Model between SOA Domain and Method Fragment (only for SOA Domain "Modelling")**

In order to better illustrate the relationship model presented in the meta-model, a concrete example is presented and explained in the next section.

### 4.1.2. Concrete example for relationship between SOA Domain Model and Method Engineering

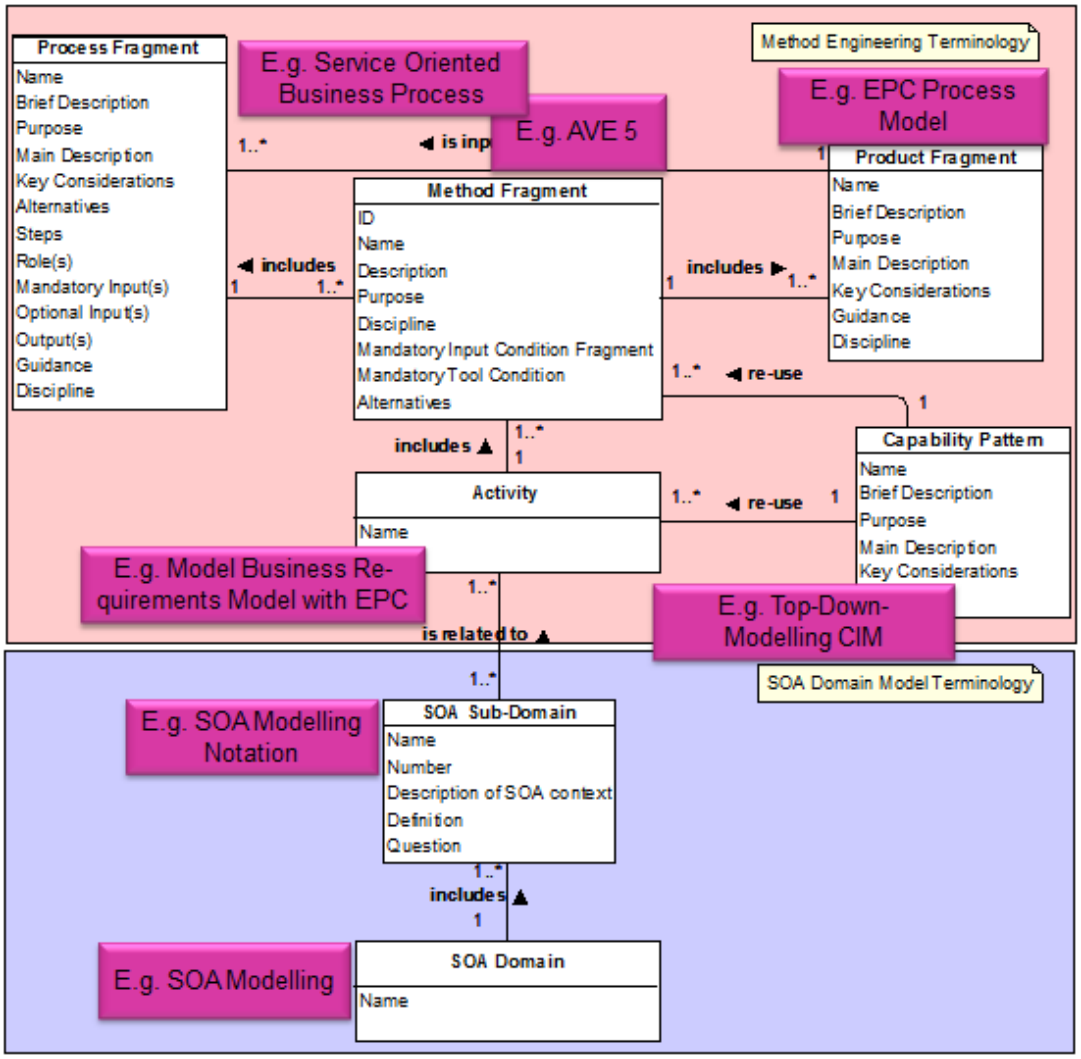



**Figure 43: Example of Alignment Model use**

The class SOA Domain "Modelling" is containing the sub-domain "1.1 SOA Modeling Notation". This sub-domain is defined, further described in the SOA context and related questions are raised: What is to be modelled on what level of detail? For the different levels of detail or abstraction layers, different activities can be found to resolve the question. Such activities are concretely "Define SOA Strategy Model", "Define CIM Model", "Define CIM2PIM Model", "Define PIM Model" etc. Each activity includes one or more method fragments: e.g. the activity "Define CIM Model" can be resolved with 3 available fragments, which are: SAP2 Services Modelling, AVE4 Enterprise Process Map and AVE5 Service oriented business Process. Every of these 3 method fragments is including "Process Fragments" and "Product Fragments". For example the fragment "AVE5 Service Oriented Business Process" includes the process fragment with details on how to apply and realize the process fragment. Conditions such as mandatory input or mandatory tools are indicated and

can influence the decision. If the conditions are fine, in this case IDS Scheers' Modelling tool is required. The details on the process fragment are important, because the context of applying this process fragment is explained and alternate options are mentioned. For instance instead of using AVE5, also SAP2 using BPMN product fragment could be used. The process fragment output will be a product fragment, in this case an EPC-Model. If this seems not to be satisfactory, a new method fragment could be formalized including e.g. UML Activity Diagram.

In order to facilitate re-use, capability patterns can be used to increase efficiency. For instance the capability pattern "SOA Strategy" is including the activities "Define SOA Strategy" and "Model Strategy". These two activities are linked to more than just one sub-domain and consequently also to one or more method fragments.

The explanation of figure 43 is now further detailed by re-using the tables from section 2.6.3. The attribute is now filled with the concrete name and descriptions are detailing the concrete examples:

**Table 26: Attributes of SOA Domain "SOA Modelling"**

| Class | Attribute | Example |
|---|---|---|
| SOA Domain | Name | SOA Modeling |

**Table 27: Attributes of SOA Sub-Domain "SOA Modelling Notation"**

| Class | Attribute | Example |
|---|---|---|
| SOA Sub-Domain | Name | SOA Modeling Notation |
| SOA Sub-Domain | Number | 1.1. |
| SOA Sub-Domain | Definition | Refer to definition 3.1.3. |
| SOA Sub-Domain | Description of SOA context | Refer to definition 3.1.3. |
| SOA Sub-Domain | Question | What are suited modeling notation candidates for the specific purpose of SOA implementation on each level of abstraction? |

**Table 28: Attributes of Capability Pattern Class**

| Class | Attribute | Example |
|---|---|---|
| Capability Pattern | Name | Top Down Modeling CIM |
| Capability Pattern | Brief Description | This pattern is describing within the CIM abstraction level the different modeling activities following a top-down approach. |
| Capability Pattern | Purpose | The purpose consists in re-using a well working set of activities re-using AVE and SAP SOA Methods on the CIM level. |
| Capability Pattern | Main Description | This pattern is consisting of 10 activities starting with the higher level of modeling and ending with the preparation to the PIM level transfer. As only AVE and SAP fragments are formalized, 8 fragments are available. |

| | | |
|---|---|---|
| Capability Pattern | Key Considerations | This pattern might be considered if business modeling is required, if top-down implementation method is selected. |
| Capability Pattern | Alternatives | Alternate patterns are Bottom-up-CIM and Meet-in-the-middle-CIM |

**Table 29: Attributes of Activity "Model Business Requirements with EPC Model"**

| Class | Attribute | Example |
|---|---|---|
| Activity | Name | Model Business Requirements with EPC Model |

The activity here "Model Business Requirements with EPC Model" is one of the activities from the capability pattern "top-down modeling CIM". In the present case, the activity is including one fragment (AVE5) but could include some more if formalized and available.

**Table 30: Attributes for Method Fragment "AVE5 Service-oriented business process"**

| Class | Attribute | Example |
|---|---|---|
| Method Fragment | ID | AVE5 |
| Method Fragment | Name | AVE5 Service-oriented business process |
| Method Fragment | Description | EPC is the standard modeling notation of ARIS, IDS Scheer tool to represent business process content. Events are triggering activities, which are performed by positions. These activities are supported by applications and data is used as in&output. An activity has one or more results (events). These activities are supported by web-services, which can be modeled related to the activity. |
| Method Fragment | Purpose | Purpose is to model business requirements on CIM level. |
| Method Fragment | Discipline | CIM |
| Method Fragment | Mandatory Input Condition Fragment | none |
| Method Fragment | Mandatory Tool Condition | ARIS Business Designer or any tool being able to support EPC modeling method. |
| Method Fragment | Alternatives | Instead of EPC Process Model, several alternates' solutions on CIM level could be selected: BPMN, UML Activity Diagram. |

**Table 31: Attributes of Process Fragment "AVE5 Service-oriented business process"**

| Class | Attribute | Example |
|---|---|---|
| Process Fragment | Name | AVE5 Service-oriented business process |
| Process Fragment | Brief Description | This process describes how to create an EPC Process Model |
| Process Fragment | Purpose | Purpose is to model business requirements on CIM |

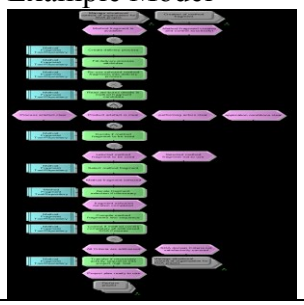
| | | |
|---|---|---|
| | | level. |
| Process Fragment | Main Description | EPC is the standard modeling notation of ARIS, IDS Scheer tool to represent business process content. Events are triggering activities, which are performed by positions. These activities are supported by applications and data is used as in output. An activity has one or more results (events). These activities are supported by web-services, which can be modeled related to the activity. |
| Process Fragment | Key Considerations | If a transformation into BPEL is foreseen, it is mandatory to follow modeling rules to enable the transformation mapping rules to be applied in a semi/automatic way. |
| Process Fragment | Alternatives | Instead of EPC Process Model, several alternates' solutions on CIM level could be selected: e.g. BPMN, UML Activity Diagram. |
| Process Fragment | Steps | Create and name EPC Model.<br>Create and name trigger event for Activity.<br>Create and name activity.<br>Create and name result event for activity.<br>Create and name position for activity<br>Create and name IT application support for activity<br>Create and name data for activity input<br>Create and name data for activity output<br>Create XOR operator rule for exclusive business decisions<br>Create OR operator rule for 1 one more business decisions<br>Create AND operator rule for parallel business logic processing<br>Create and name end event |
| Process Fragment | Role(s) | Business Analyst |
| Process Fragment | Mandatory Input(s) | none |
| Process Fragment | Optional Input(s) | none |
| Process Fragment | Output(s) | EPC Process Model |
| Process Fragment | Guidance | none |
| Process Fragment | Discipline | CIM |

**Table 32: Attribute of Product Fragment "EPC Process Model"**

| Class | Attribute | Example |
|---|---|---|
| Product Fragment | Name | EPC Process Model |
| Product Fragment | Brief Description | The EPC Process Model is a process notation language to represent business requirements. The process flow and sequence is showed with events and activities. Additionally, information can be modeled on who |

| | | |
|---|---|---|
| | | (Roles) is performing activities, with what application the activity is supported, what data is used in/out of an activity. |
| Product Fragment | Purpose | EPC Process Models is used by many companies for modeling, analyzing, and redesigning business processes. As such it forms the core technique for modeling in ARIS, which serves to link the different views in the so-called control view, which will be elaborated in section of ARIS Business Process Modeling. |
| Product Fragment | Main Description | Event: Events are passive elements in EPC. They describe under what circumstances a function or a process works or which state a function or a process results in. Examples of events are "requirement captured", "material on stock", etc. In the EPC graph an event is represented as hexagon. In general, an EPC diagram must start with an event and end with an event.<br>Function: Functions are active elements in EPC. They model the tasks or activities within the company. Functions describe transformations from an initial state to a resulting state. In case different resulting states can occur, the selection of the respective resulting state can be modeled explicitly as a decision function using logical connectors. Functions can be refined into another EPC. In this case it is called hierarchical function. Examples of functions are "capture requirement", "check material on stock", etc. In the EPC graph a function is represented as rounded rectangle.<br>Organization unit: Organization units determine which person or organization within the structure of an enterprise is responsible for a specific function. Examples are "sales department", "sales manager", "procurement manager", etc. It is represented as an ellipse with a vertical line.<br>Information, material, or resource object: In the EPC, the information, material, or resource objects portray objects in the real world, for example business objects, entities, etc., which can be input data serving as the basis for a function, or output data produced by a function. Examples are "material", "order", etc. In the EPC graph such an object is represented as rectangle.<br>Logical connector: In the EPC the logical relationships between elements in the control flow, that is, events and functions are described by logical connectors. With the help of logical connectors it is possible to split the control flow from one flow to two or more flows and to synchronize the control flow from two or more flows to one flow.<br>If function F1 completes,<br>either events E1 or E2 occur If either events E1 or E2 occur, function F1 starts<br>Logical relationships |

| | | |
|---|---|---|
| | | There are three kinds of logical relationships defined in EPC:<br><br>1.) Branch/Merge : Branch and merge correspond to making decision of which path to choose among several control flows. A branch may have one incoming control flow and two or more outgoing control flows. When the condition is fulfilled, a branch activates exactly only one of the outgoing control flows and deactivates the others. The counterpart of a branch is a merge. A merge may have two or more incoming flows and one outgoing control flow. A merge synchronizes an activated and the deactivated alternatives. The control will then be passed to the next element after the merge. A branch in the EPC is represented by an opening XOR, whereas a merge is represented as a closing XOR connectors.<br><br>2.) Fork/Join : Fork and join correspond to activating all paths in the control flow concurrently. A fork may have one incoming control flow and two or more outgoing control flows. When the condition is fulfilled, a fork activates all of the outgoing control flows in parallel. A join may have two or more incoming control flows and one outgoing control flow. A join synchronizes all activated incoming control flows. In the EPC diagram how the concurrency achieved is not a matter. In reality the concurrency can be achieved by true parallelism or by virtual concurrency achieved by interleaving. A fork in the EPC is represented by an opening 'AND', whereas a join is represented as a closing 'AND' connectors.<br><br>3.) OR : An 'OR' relationship corresponds to activating one or more paths among control flows. An opening 'OR' connector may have one incoming control flow and two or more outgoing control flows. When the condition is fulfilled, an opening 'OR' connector activates one or more control flows and deactivates the rest of them. The counterpart of this is the closing 'OR' connector. When at least one of the incoming control flows is activated, the closing 'OR' connector will pass the control to the next element after it.<br><br>Control flow: A control flow connects events with functions, process paths, or logical connectors creating chronological sequence and logical interdependencies between them. A control flow is represented as a dashed arrow.<br>Information flow: Information flows show the connection between functions and input or output data, upon which the function reads changes or writes. |

| | | |
|---|---|---|
| | | Organization unit assignment: Organization unit assignments show the connection between an organization unit and the function it is responsible for.<br><br>Process path: Process paths serve as navigation aid in the EPC. They show the connection from or to other processes. The process path is represented as a compound symbol composed of a function symbol superimposed upon an event symbol. To employ the process path symbol in an EPC diagram, a symbol is connected to the process path symbol, indicating that the process diagramed incorporates the entirety of a second process which, for diagramatic simplicity, is represented by a single symbol. |
| Product Fragment | Key Considerations | This Process Model Notation on CIM Layer needs to be considered if the process design tool integrated EPC notation. Generally, it is difficult to execute or transfer into an execution environment. This notation can be used, if ARIS SOA Architect as tool is foreseen with a later transformation into BPEL notation. A semi-automatic transformation CIM2PIM is available in ARIS SOA Architect. |
| Product Fragment | Guidance | Example Model<br>  |
| Product Fragment | Discipline | CIM |

### 4.1.3. Engineering Method for SOA Implementation

The following section will define and formalize the method to follow for a SOA engineering method. This method is referring as described earlier to the definition of method from Vernadat [Ver96] which is "a (1) set of methods, (2) models and (3) tools to be used in a structured way to solve a problem." The SOA engineering method is a set of processes, which are realized or performed with the help of tools (method fragment creation, situational assessment, selection&assembly of fragments). The facilitation by tools will be described in detail in the "Tooling & Prototyping" chapter 5. We consider that either the project manager or a method engineer is performing or executing the processes as described. The process of describing the creation of SOA Domains and SOA Sub-Domains is available but not formalized and described in detail here. The description of attributes and relationship between both classes were introduced in section 3.1.1. As method, we use as already mentioned situational ME.

#### 4.1.3.1. Engineering Process for SOA implementation

The application process overview is containing 5 different processes. The definition of these processes has been inspired by the ME-processes as illustrated by Mayer et al. [MCFKP+95] (figure 23) and Brinkkemper [Bri96] (figure 24). These 5 proposed processes are the following:

- Creation of method fragment
- Manage situational context of organization for SOA project
- Selection of method fragment & assembly of method fragments
- Perform project
- Update method fragments after project end

As usual for processes, it is interesting to show these 5 processes through fragment definition, method design and method application:

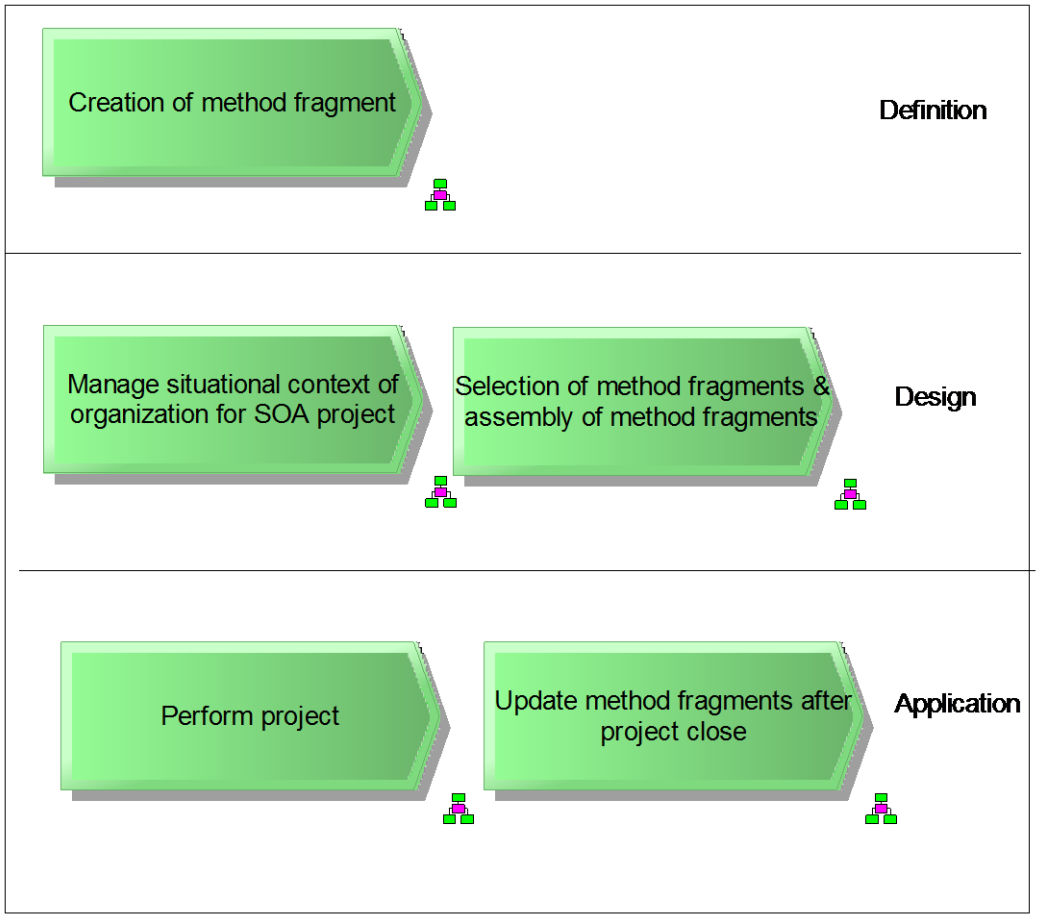



**Figure 44: Engineering Process for SOA implementation Workflow View**

For the formalization of these 5 processes EPC modelling is applied. The following objects are used:

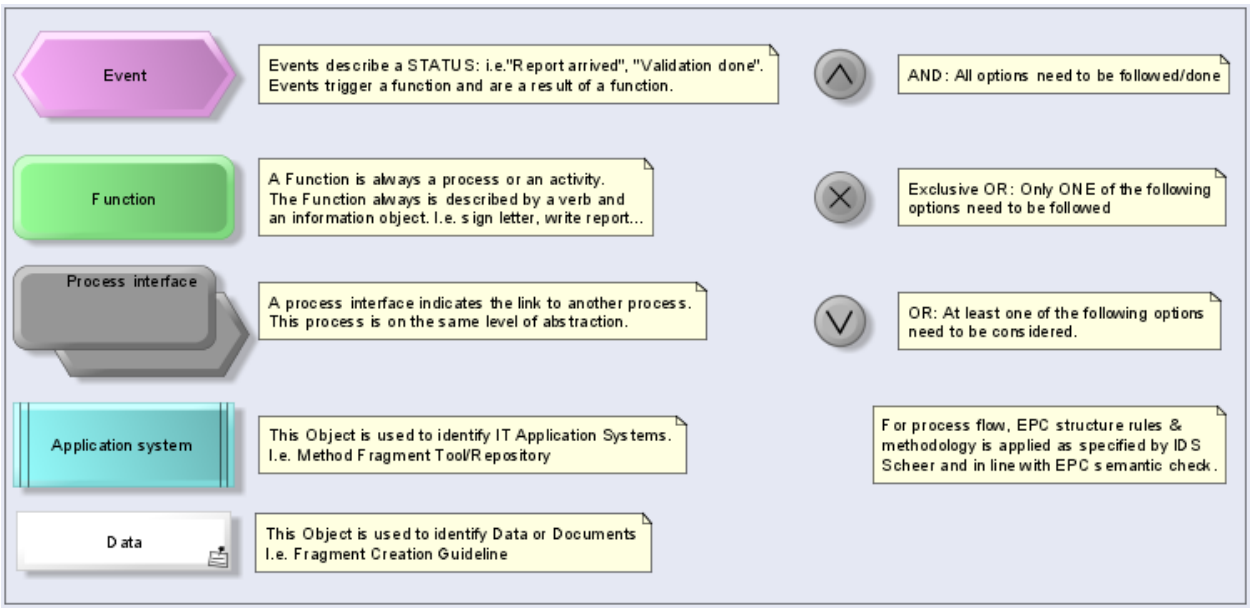



**Figure 45: Legend for SOA Engineering Process Models**

The next sections will detail the 5 processes re-using Event-Driven-Process Chains (EPC) method following the legend description in figure 45. To ensure object-oriented modelling, and the usage of object re-use, the link between dynamic (processes) and static (application, system and data) views is the following:

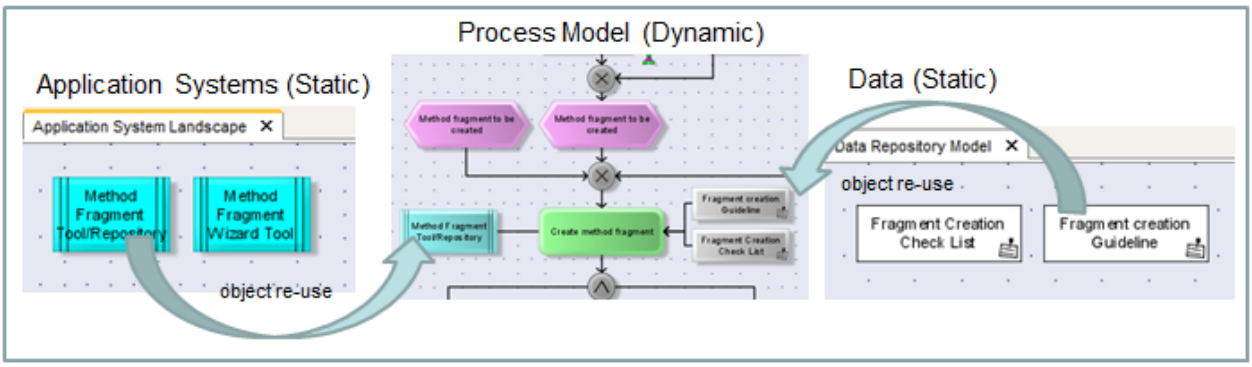



**Figure 46: Object Re-Use between Static and Dynamic Views**

A detailed report on object information is provided on the accompanying CD. The object information explains exactly where objects are re-used and which connections they have to other objects.

### 4.1.3.2. Creation of Method Fragment

This process has as an objective to populate the method fragment database. This prerequisite is necessary as input to allow the availability of method fragments in the fragment database.

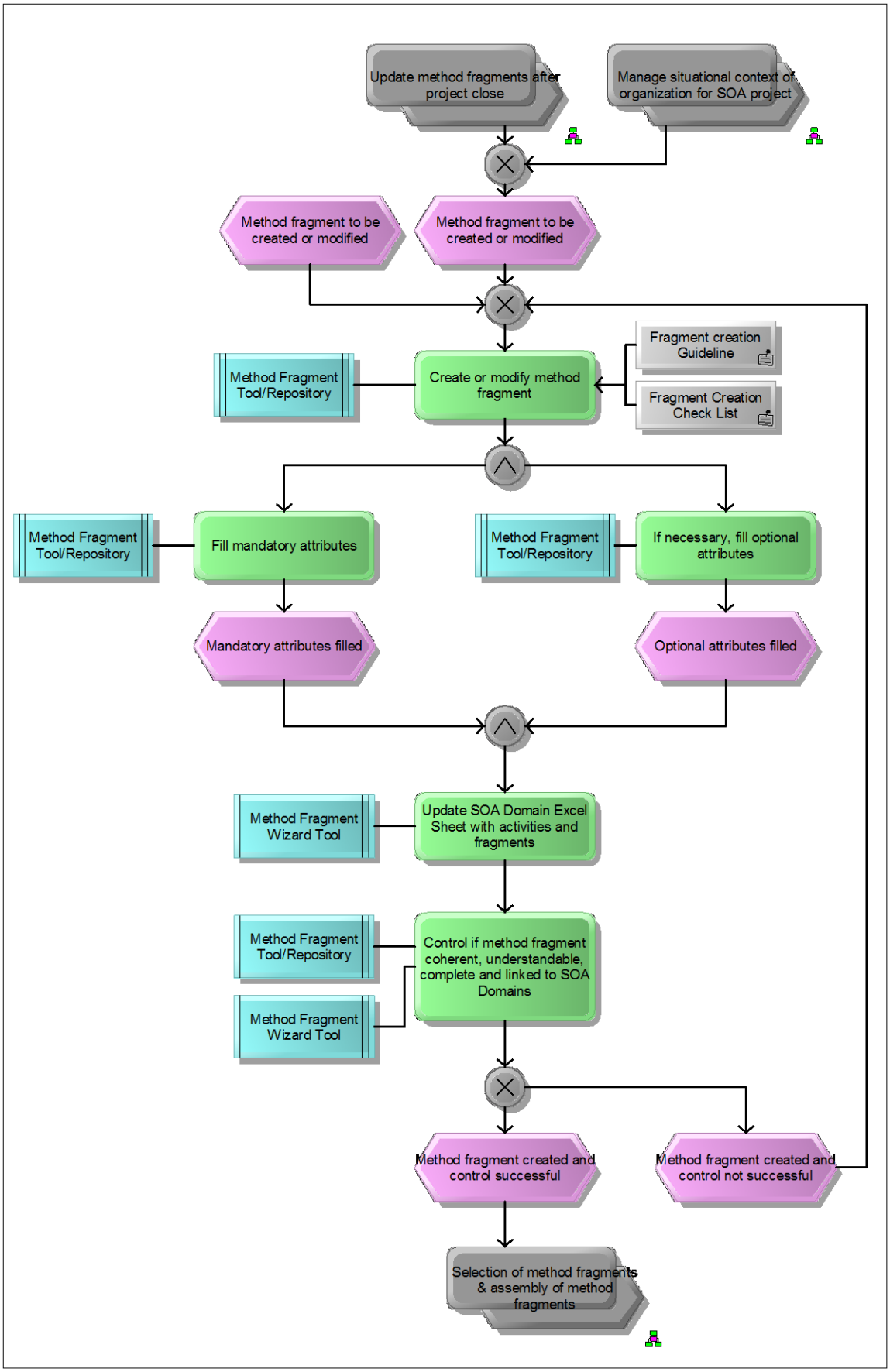



**Figure 47: Process: Create Method Fragment**

The start of this project can be either triggered from generic method fragments to be created or from the update of available method fragments after project closure. It could also be that during the situational context selection, method fragments are missing in the database and therefore need to be created. The method fragment includes the process fragment and the product fragment. The method fragment is created in the EPF tool which is integrating SPEM2.0. method. A checklist is supporting the activity of filling the available attributes accordingly. Mandatory and optional fields are indicated. After completing the checklist and the new fragments are available, an update in the excel SOA Domain tool needs to be done. This Excel file is a facilitating sheet, which is further described in section 5.4. An additional control at the end ensures coherence, completeness and understandability of fragments.

#### 4.1.3.3. Manage Situational context of Organization for SOA project

This process has as objective to capture the situational context of the organization with the SOA domains developed in this thesis.

The SOA Domains are containing the sub-domains defined earlier. Each sub-domain definition as well as the SOA contextual issues needs to be understood. Based on this, priorities are set and considered for the decision on how to address the criteria's. Organization specific context needs to be gathered; similar to the field trial application examples in chapter 6 (e.g. section 6.3.2. and 6.3.7. for details). Based on organization specific content, priorities must be decided on e.g. what implementation strategy to use, what systems to use, how big the scope of the project is etc. Based on these organization priorities, the generic activities can be selected. As between the activities chosen, there is a link to the SOA sub-domains, the method user can decide if each sub-domain is sufficiently addressed by activities or if eventually some risks should be taken by non-addressing. If the sub-domain is not sufficiently covered and the risk estimation is too high, it needs to be evaluated if method fragments are available and also meeting requirements. If a method fragment is not available, the process executer needs to decide if this fragment has to be created or not. If the creation is not an option, the process loops back to the decision on SOA sub-domain coverage. If a method fragment is available in the fragment database, the process continues with the selection of method fragments.

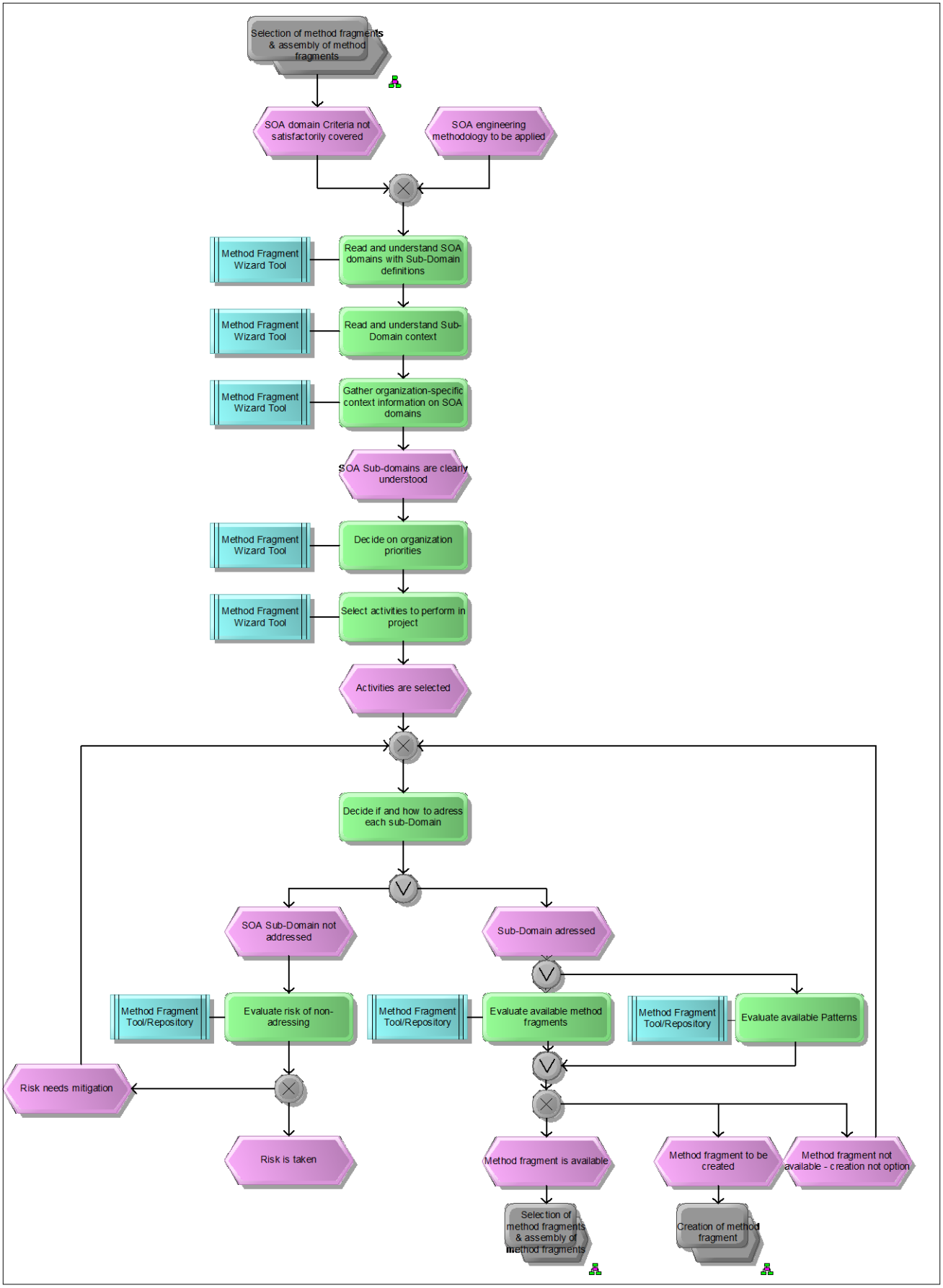



**Figure 48: Process Manage situational context of organization for SOA project**

#### 4.1.3.4. Selection & Assembly of Method Fragments

This process has as objective to select the fitting method fragment to the situation identified in the earlier process.

First, a delivery process has to be created for the individual project. The delivery process attributes such as “name”, “description”, “purpose” and “scope”. Next this delivery process needs to be populated with fragments. For this, fragment candidates are checked in detail in the method fragment tool. Based on the attribute descriptions of method fragments, the method applicant has to judge if the fragment candidate is still a good choice. Here, several important information are made clear: the product outcome of the fragment, the process steps, the actors performing the fragment, the prerequisites or conditions related to other related fragments or tools necessary. If this is accepted, the fragment is selected.

The selected method fragments are re-used from the “method content” area into the “process” application area. The method fragments are compiled into a sequence. Then a control to identify method coverage of SOA Domains is performed. If the coverage needs to be improved, a process link refers back to the process “manage situational context of organization for SOA project“. If a project mgt. tool is used, a merge or input needs to be done to have a ready-to-use project plan:

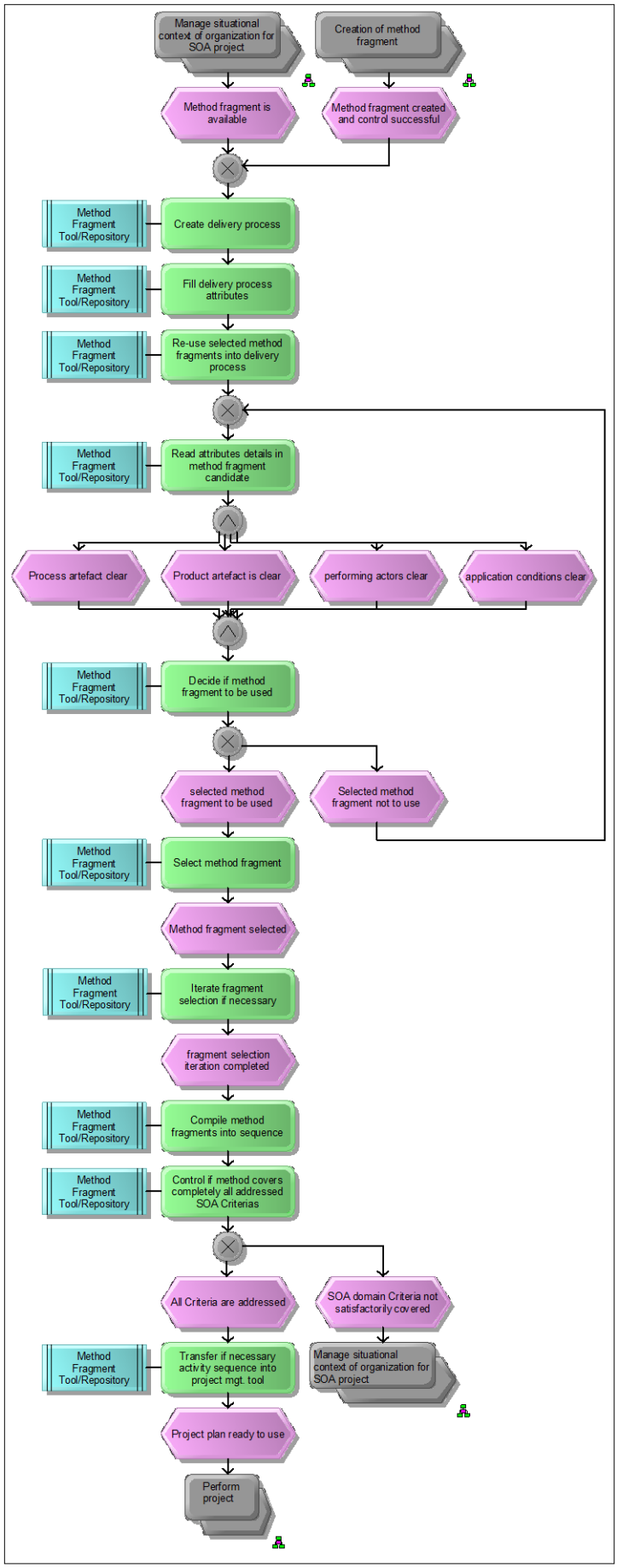



**Figure 49: Process: Selection of Method Fragment**

#### 4.1.3.5. Perform Project

The objective of this process is to describe the activity around starting the project and communicating the method to project team and stakeholders.

Once the project-plan is finalized, the approach needs to be explained to the project team and stakeholders. Method fragment tool are normally providing a functionality to create html-files to allow project team and stakeholder information and guidance along the project execution:

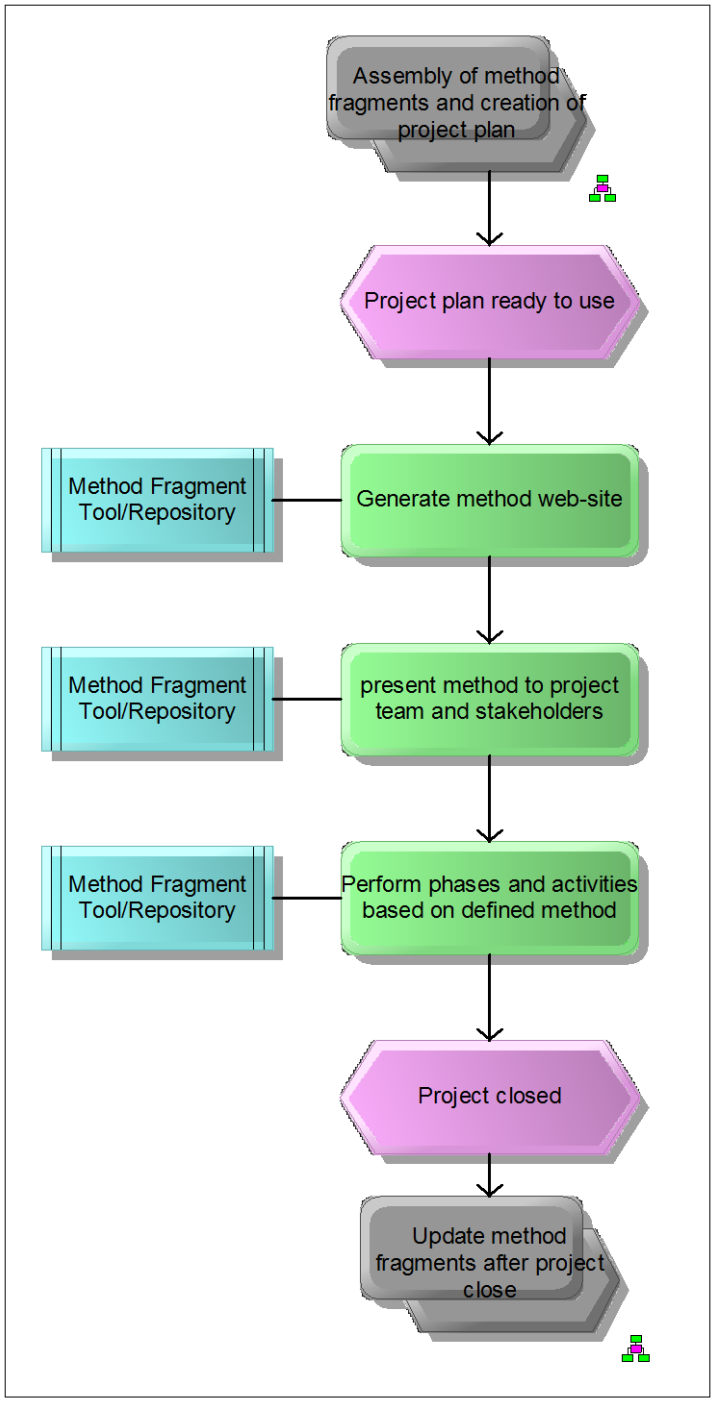



**Figure 50: Process: Perform Project**

### 4.1.3.6. Update Method Fragments after Project Close

This process has as objective to record project experience on used method fragments in order to enrich already available information or to generate new process fragments.

First, lessons learned or contextual information needs to be summarized on every applied method fragment. Available method fragments in the database are updated with additional information of project experience. Next, eventually new process fragments generation could be triggered (Interface to process: "Creation of method fragment").

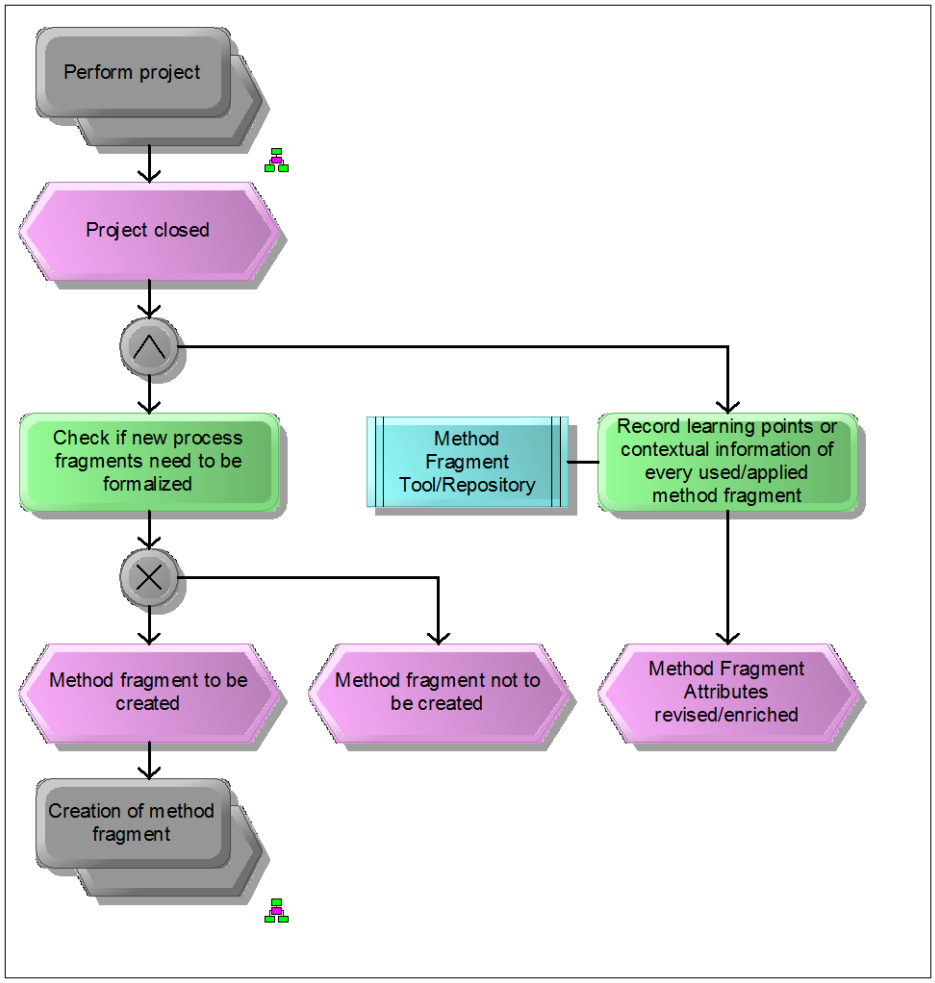



**Figure 51: Process: Update method fragments after project close**

The next section will illustrate the application of the first mentioned process "Creation of Method Fragment".

## 4.2. Artefact 4: SOA Method Fragments

### 4.2.1. Formalizing Fragments from available SOA Methods

The relevant attributes to identify and describe a fragment has been introduced in table 15.

The following work products have been defined based on fragment input/output required [WM06] [Yva06]:

- Access Diagram (Data and Ontology Model with the objective to show relationships to activities, organization, applications)
- Balanced Scorecard Model (Strategy Model integrating BSC method)
- BPEL Diagram (Technical model representing web-service orchestration)
- BPMN Process Model (Business Process Model using BPMN notation)
- EPC Process Model (Business Process Model using EPC notation)
- EPC2BPEL Transformation (Transformation Mechanism from EPC to BPEL)
- IT Strategy Document (IT Strategy description)
- KPI Allocation Diagram (Model for details on how objectives are measured with KPIs)
- Value Added Chain Process Model (Value Chain Model based on Porter)
- WSDL (Web-service description language document)

As the previous sections have shown exemplarily the detailed class descriptions (refer to table 13-17) of the alignment model (figure 43) and one concrete example (refer to table 26-32) some more method fragments have been formalized from available methods.

Because of time restrictions, we will concentrate on extracting fragments from 2 SOA methods, which is an instantiation of the configuration process for SOA situational method. The two selected SOA methods are Enterprise SOA Adoption Strategies [WM06] and ARIS Value Engineering for SOA [Yva06]. The reason for this choice consists in the fact that some of these fragments can be re-used in the field trial in chapter 6. Only these attributes defined in tables 19+20 and 26-32 have been formalized. These formalized fragments are available in the method fragment tool (refer to chapter 5).

In case that the organizations in the field trials (chapter 6) would have had other tools such as Rational Rose, the IBM method SOMA or SoDD would have been formalized instead.

A difficulty consists certainly in detailing the right level of granularity and decomposition of method fragments. To solve this, the term activity can help to indicate the right level of detail. The next tables will illustrate this decomposition and the related details to identified method fragments.

**Table 33: Formalized Method Fragments Enterprise SOA Adoption (SAP)**

| ID | SAP 1 | SAP 2 | SAP 3 |
|---|---|---|---|
| Name | Discover Vision & Opportunities | Services Modelling | Build Services |
| Description | The emphasis is on learning and understanding ESAs potential for enhancing | The fragment is detailing how to model and design process components with | The fragment is detailing how to create new web-services (from scratch) or if some pre-configured |

| | | | |
|---|---|---|---|
| | the organisations business. The aims are particularly to grasp the value of the Netweaver platform, Identify opportunities for applying the enterprise service idea and explore the TCO of this approach. | BPMN notation including the identification of Business Objects. Next, User interfaces have to be described and finally service candidates have to be determined. | web-services from a SAP web-service library can be used. The web-service is linked to a business object, which is part of the BPMN process artefact. This is done with the specific SAP development tools (Web Dynpro). The web-service is described in WSDL and written into the service registry for publishing and classifying the web services. |
| Purpose | This fragment is supposed to provide a vision of SOA and ESA (Enterprise Service Architecture) in particular. | This fragment is supposed to provide BPMN models, which can be deployed into the SAP process execution engine. | This fragment is supposed to plan, build and implement web-services into the into SAP NetWeaver execution engine. |
| Discipli ne | Strategy | CIM | PIM |
| Mandat ory Input Con. Frag. | None | None | None |
| Mandat ory Tool Con. | None | SAP PI and NetWeaver | SAP PI and NetWeaver |
| Alter-natives | AVE1 | None | None |

Further SAP fragments were not formalized as these were more addressing other SOA Domains than the SOA Domain Modelling.

**Table 34: Method Fragments ARIS Value Engineering for SOA (IDS Scheer)**

| ID | AVE 1 | AVE 2 | AVE 3 | AVE 4 | AVE 5 |
|---|---|---|---|---|---|
| Name | Envision Service Architecture Management | Business Goals with Balanced Scorecard | Detail Key Performance Indicators | Enterprise Process Map | Service Oriented Business Process |
| Des-cripton | Within the IT objectives, it should be explained WHY exactly SOA | In the cause-and-effect diagram of the Balanced | In the KPI allocation diagram for a Balanced Scorecard, strategically | The value-added chain diagram is mainly used to identify the functions | EPC is the standard modelling notation of ARIS, IDS Scheer tool to |

| | | | | | |
|---|---|---|---|---|---|
| | is required and what the expected benefits are. The project should be positioned in the wider scope of IT strategy direction. | Scorecard (BSC), the necessary objectives and critical factors for implementing a business strategy are defined and their mutual influence is depicted using a cause-and-effect chain running over perspectives. | relevant objectives or crucial critical factors can be assigned both the KPIs for evaluating the achievement of objectives and initiatives to be performed. | within a company that are directly involved in the creation of a company's value added. These functions can be interlinked as a sequence of functions and thus form a value-added chain. | represent business process content. Events are triggering activities, which are performed by positions. These activities are supported by applications and data is used as in&output. An activity has one or more results (events). These activities are supported by web-services, which can be modeled related to the activity. |
| Purpose | This fragment is supposed to identify strategic business objectives and derive IT objectives from that. | The purpose of this fragment is to derive from Organization goals and drivers SOA objectives. | The purpose of this fragment is model strategically relevant objectives or crucial critical factors which can be assigned to the KPIs for evaluating the achievement of objectives and initiatives to be performed. | Get an understanding of value added processes landscape. | Purpose is to model business requirements on CIM level. |
| Discipline | Strategy | Strategy | Strategy | CIM | CIM |
| Mandatory Input Cond. Fragm. | None | None | None | None | None |
| Mandatory Tool Cond. | None | ARIS Business Architect incl. BSC Extension | ARIS Business Architect with BSC Extension | ARIS Business Architect with BSC Extension | ARIS Business Architect |
| Alternatives | SAP 1 | None | None | None | SAP 2 |

As already introduced in chapter 2, an important term bridging the class sub-domain and method fragment is used. This term is needed to allow a classification of fragments on the same level of abstraction and to "neutralise" the fragment from the source semantics. Therefore, the SPEM/EPF term "activity" is used. One activity e.g. "Define SOA Strategy" can be materialized by two different fragments:

1.) "Envision Service Architecture Management" which is the AVE1 fragment and
2.) "Discover Vision and Opportunities" which is the SAP1 fragment.

Both have the same objective to represent the strategic consideration for SOA in natural language. The difference of both fragments is the source of method provider and the possible connection to other fragments. The SAP1 one is more a stand-alone and has no direct link to the next fragment SAP2. The AVE is much more focussing on strategy and also on how to represent this strategy in models (AVE2, AVE3) and a link is made into the Enterprise Process Map (VACD Diagram), which does not exist in SAP method.These connections are described in the attributes of the method fragment description.

Again, activities are needed to provide a generic activity basket, which is including one or more process fragments among which to select.

The following table is classifying the fragments by abstraction layer and by additionally indicating the activities from SOA sub-domain Modelling:

**Table 35: Method Fragments Summary**

| Abstraction Layer | Activity | Method Fragment | Work Product |
|---|---|---|---|
| Strategy | Define SOA Strategy | AVE1 Envision Service Architecture | Natural Language Document |
| Strategy | Define SOA Strategy | SAP1 Discover Vision and Opportunities | Natural Language Document |
| Strategy | Model Strategy with BSC | AVE2 Business Goals with Balanced Scorecard | BSC Model |
| Strategy | Model Strategy Key Performance Indicators | AVE3 Detail Key Performance Indicators | KPI Allocation Diagram |
| CIM | Model Value Chain For Process Overview | AVE4 Enterprise Process Map | VACD Diagram |
| CIM | Model Business Requirements with BPMN | SAP2 Services Modeling with BPMN | BPMN Model |
| CIM | Model Business Requirements with EPC | AVE5 Service Oriented Business Process | EPC Model |
| CIM | Model Technical BPMN | SAP3 Build Services | BPMN Model |

The benefit of this consists in the capability to select one of the different fragments available in the database for a concrete delivery process.

## 4.3. Summary on Research Contribution

First, the SOA Domain Model has been constructed based on the input from the state-of-the-art on available SOA methods, SOA modeling candidate notations, model interfacing mechanisms etc. The construction process of the Domain Model has been explained as well as the details of each SOA Domain with its sub-domains (Artifact 1).

Available SOA methods have been analyzed and qualified against the SOA Domain model. The result was that none of the available methods is covering all SOA Domains as explained in table 25. This result confirms the research gap as introduced in the first chapter (Artifact 2).

A worldwide questionnaire should test the created SOA Domain Model with the related sub-domains. 90,74% of respondents agreed on completeness of the SOA Domain Model. Furthermore, knowledge was gathered on industrial basis to complete and fine-grained desk research and state-of-the-art.

In order to link the identified sub-domains with an engineering method, ME principles have been used. The SOA Domain Model has been aligned with ME terminology to allow common understanding of the concept. Each class of the model has been explained and examples were given. Exemplarily, the alignment model was only applied for the SOA Domain "Modeling".

Next, five configuration processes for SOA situational methods were created. These processes explain how to create process fragments, how to apply them in a situational method, how to assemble and select these fragments, how to perform the project and finally how to update method fragments after project experience (Artifact 3).

For demonstration of feasibility, several method fragments have been created from available SOA methods. These fragments have been formalized by describing the attributes and also by providing detailed examples of this formalization (Artifact 4).

# CHAPTER 5

# PROTOTYPING OF A TOOLING SUPPORT FOR SOA METHOD ENGINEERING FRAMEWORK



For the instantiation of method fragments (refer to table 1), it is required to prototype a tooling support for the SOA Engineering Framework aligning with method engineering terminology (refer to table 37). Chapter 5 is about formalizing research contribution artefacts from chapters 3 and 4 into an applicable and structured prototyping of a tooling support for SOA method engineering framework. An introduction on used tools and produced artefacts is given (5.1.). The second section (5.2.) is introducing the framework tooling allowing an overview on the main outcomes such as the SOA Domain Model, SOA Alignment Model and the SOA Engineering Process. For formalizing and structuring method fragments, the Eclipse Process Framework (EPF) Tool is used. Section (5.3.) is detailing on how available method content is formalized with EPF Tool. Next, an Excel-based facilitating guideline to apply the SOA Domain Model in a situational context with its content is explained (5.4.). The chapter concludes with summarizing tooling and prototyping experience (5.5.).

## 5.1. Introduction to Prototyping of a Tooling Support for SOA Method Engineering Framework

The tools have mainly two objectives: First, the user (all persons who have an interest in this work) should have a user-friendly framework to allow understanding of SOA Engineering Method components. The tool should easily present the framework with the main artefacts.

Second objective is the enforcement of ME principles, which are necessary as situational ME has been chosen to propose a solution that can cope to different situations. In order to do this in an efficient way, SPEM2.0. has been chosen to fulfil the requirements of ME on formalized language and modelling. To enforce this language with the rules, a tool was used to manage the complexity and implement the ME requirements. For this, EPF Composer version 1.5.0.4. has been used. Semantics between the SOA Domain Model, SPEM 2.0. and EPF terminologies have been aligned to ensure common understanding.

A modelling tool has been used to create artefacts and a decision aid file under Excel format supports fragment selection process:

**Table 36: Overview on used tools and produced artefacts**

| Tool Type | Tool Used | Produced Tool Artefact |
|---|---|---|
| Modelling Tool | ARIS SOA Architect | Process Models (VACD, EPC), Alignment Diagram (UML Class) Object-Oriented Database for Processes, Web-Portal for navigation through SOA Framework |
| Method Tool | Eclipse Process Framework | Method Fragments, Fragment Database |
| Decision Aid Tool | MS Excel | Decision table for fragment selection |

All three tools with artefacts are available on the CD-ROM supporting this PhD document.

## 5.2. SOA Engineering Framework with Modeling Tool

The SOA Framework modelling Tool with the html generator is the single point of entry for the overview of the produced 4 research artefacts as introduced in chapter 1.

The framework as shown in figure 52 is available for users in HTML-format, meaning that the complete content can be browsed through. This way, researchers or practitioners get an easy-to-use tool, which is concentrating and summarizing the main contributions. Furthermore, the framework is easy to share and distribute as the content can be published through web or intranet sites. The user of this framework tool can drill-down into the following artefacts: SOA Domain Model (section 3.1.) and SOA Alignment Model (section 4.1.1.), the SOA method qualification results (3.2.), the configuration process for SOA situational method (section 4.1.) and the list of available method fragments in EPF (section 4.2.):

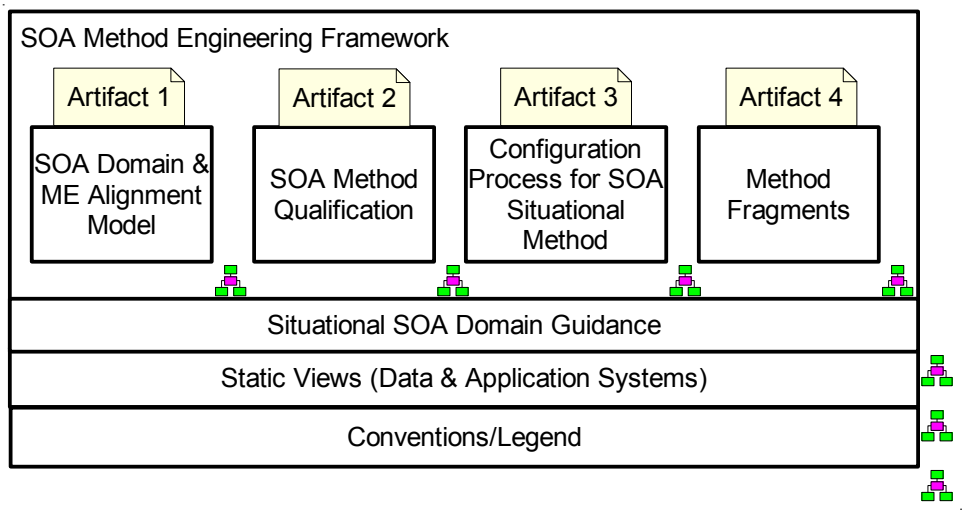



**Figure 52: SOA Engineering Method Framework (Screenshot)**

This facilitating view is containing 7 building blocks, which are the SOA Domain Model (refer to table 21) and the alignment model between SOA Domain Model and ME alignment

(refer to figure 46), the SOA Method Qualification table (refer to table 25), the Configuration Process for SOA Situational Method (refer to figures 47-51), and a list of method fragments (refer to table 33-34). Furthermore, a link to the SOA domain guidance Excel tool (section 5.3.) is available as well as a static repository on data, application & systems, (refer to figure 46). Next, a conventions/legend as guidance is available on how to read processes (refer to figure 45).

The presented tooling approach is aligned with the definition of "method" as we have created a method to be "a (1) set of methods, (2) models and (3) tools to be used in a structured way to solve a problem." (refer to definition 32, Vernadat [Ver92]).

## 5.3. Method Fragment Formalization with Eclipse Process Framework (EPF) Tool

The objective of this section is to demonstrate the application of available methods and to show the ability of formalizing them with ME. As introduced in chapter 1, the tooling should support the validation of ideas and support the field trial studies in chapter 6. In order to restrict the scope to a meaningful size, only the SOA Domain "Modelling" with its "sub-domains" has been formalized in fragments (refer to section 4.2.). It is important to demonstrate that the conceptual foundation in chapter 3 and 4 can be implemented with tools. All definitions related to concepts in that specific part of the method with presented artefacts have been carefully gathered and applied in the ME context using SPEM 2.0. and EPF [Eclipse09] to formalize the method. The alignment model to bridge conceptually the SOA Domains and the method fragments has been introduced and explained in chapter 4 (refer to section 4.1. and 4.2.).

As mentioned earlier, the formalized methods into fragments are taken out of documentation provided by IDS Scheer [Yva06] (refer to section 3.2.2.) and SAP [WM06] (refer to section 3.2.3.). The SOA Domain Model sub-domains have been introduced as "customized categories" (EPF tool terminology) into the tool and are linked to the fragments. Each process fragment is formalized with name, description and purpose of the fragment. The fragment is further detailed by steps. Work products are indicated and can be distinguished in mandatory input, optional input and output. Rules have been added by defining mandatory or optional fragments, input/output relationships on work products and predecessors in a work-breakdown structure (Refer to table 33 and 34) e.g. as an example: "EPC Modelling is mandatory for EPC2BPEL" fragment. Guidance is giving an illustration or helping information to the fragment. Within the category selection, we distinguish two criteria's: the disciplines (EPF tool terminology) are indicating the abstraction level (Strategy, CIM, PIM, PSM) and the customized category is indicating the relationship to the SOA Domain Model criteria's. One or more criteria's can be selected. Additionally, the term alignment table is enriched with semantics that EPF tool is using:

**Table 37: Terminology alignment table between SPEM2.0. and EPF Tool**

| SPEM 2.0. Semantic/ Definition | Eclipse Process Framework Tool |
|---|---|
| Role Definition | Role |
| Task Definition | Task |
| Work Product Definition | Work Product |

| Role Use | Role |
|---|---|
| Task Use | Task Description |
| Work Product Use | Work Product |
| Step | Step |
| Description | Description |

This table ensures understanding of different terminologies and semantics used in SPEM and EPF. SPEM has been introduced in detail in section 2.6.3. and requirements for such tools have been introduced in detail in section 2.6.4. In order to illustrate the implementation in the EPF tool, figure 53 summarizes exemplarily the content and relationships of method fragment "Service Oriented Business Process EPC" between objects:

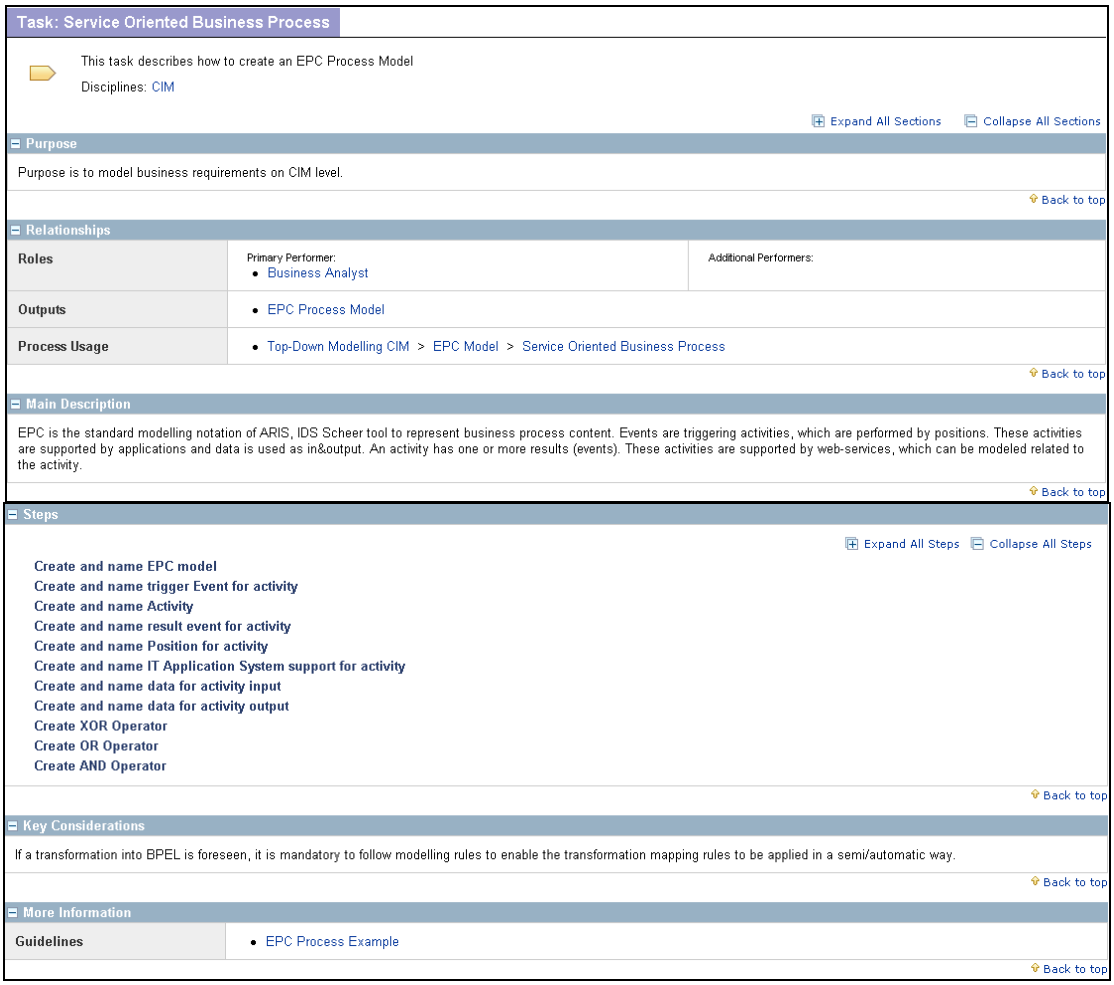



**Figure 53: Method Fragment "Service Oriented Business Process EPC" in EPF Tool (screenshot)**

The task is called Service Oriented Business Process (short name AVE5) and is categorized on CIM level. As defined in chapter 3, formalized attributes are describing the process fragment as defined in table 25. The full content of this fragment has been detailed in table 32. The relationships such as outputs (EPC Process Model) are listed. The detailed steps of this process fragment are given as well as a guideline which is in this case an EPC process example illustration.

Next, the product fragment "EPC Process Model" screenshot is given:

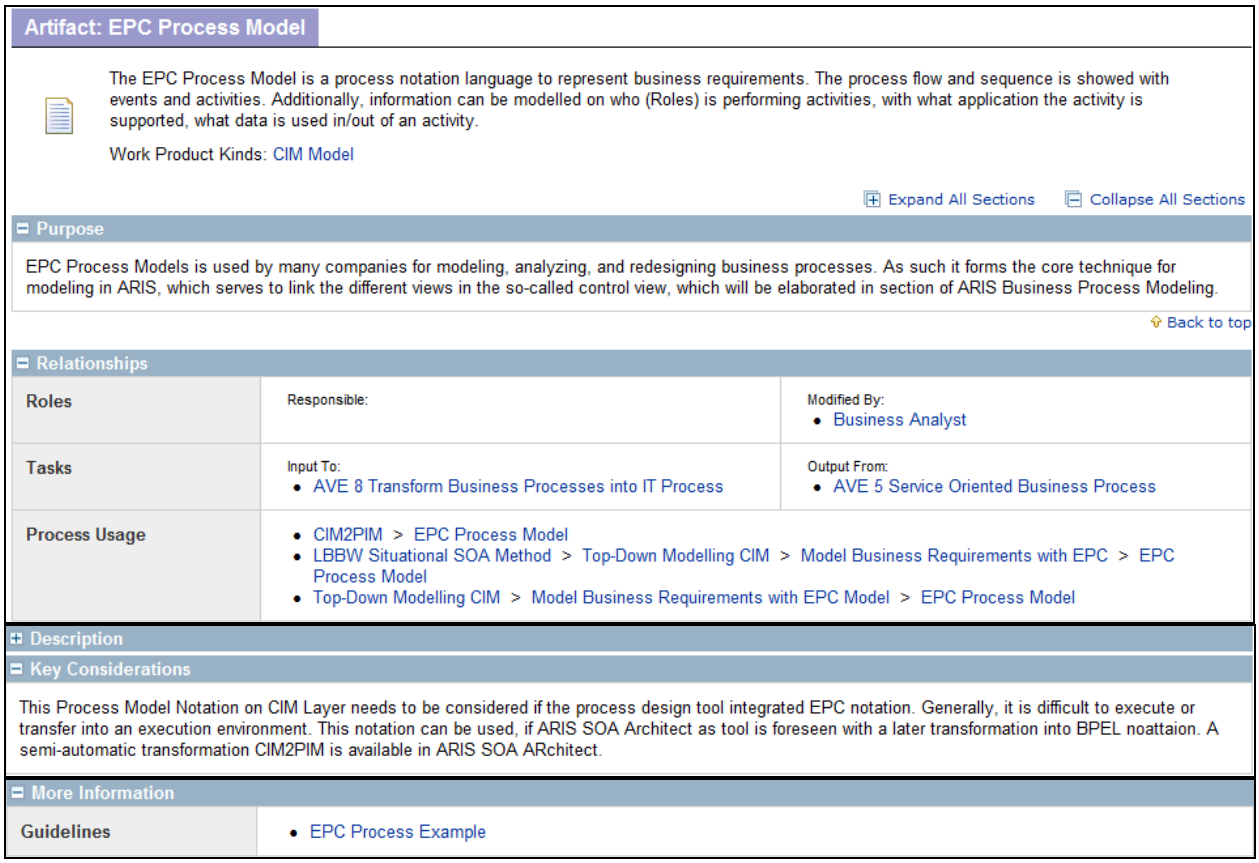



**Figure 54: Method Fragment "EPC Process Model" in EPF Tool (Screenshot)**

Similarly as for the process fragment, attributes are describing the product fragment as defined in table 18 and table 33 (full details of fragment). For space reasons, the description has not been expanded. Additionally to the defined and explained attributes, EPF automatically indicates relationships to other fragments (Input or Output) and where the product fragment is used. Again, a guideline is given with an illustration of a concrete EPC process model example.

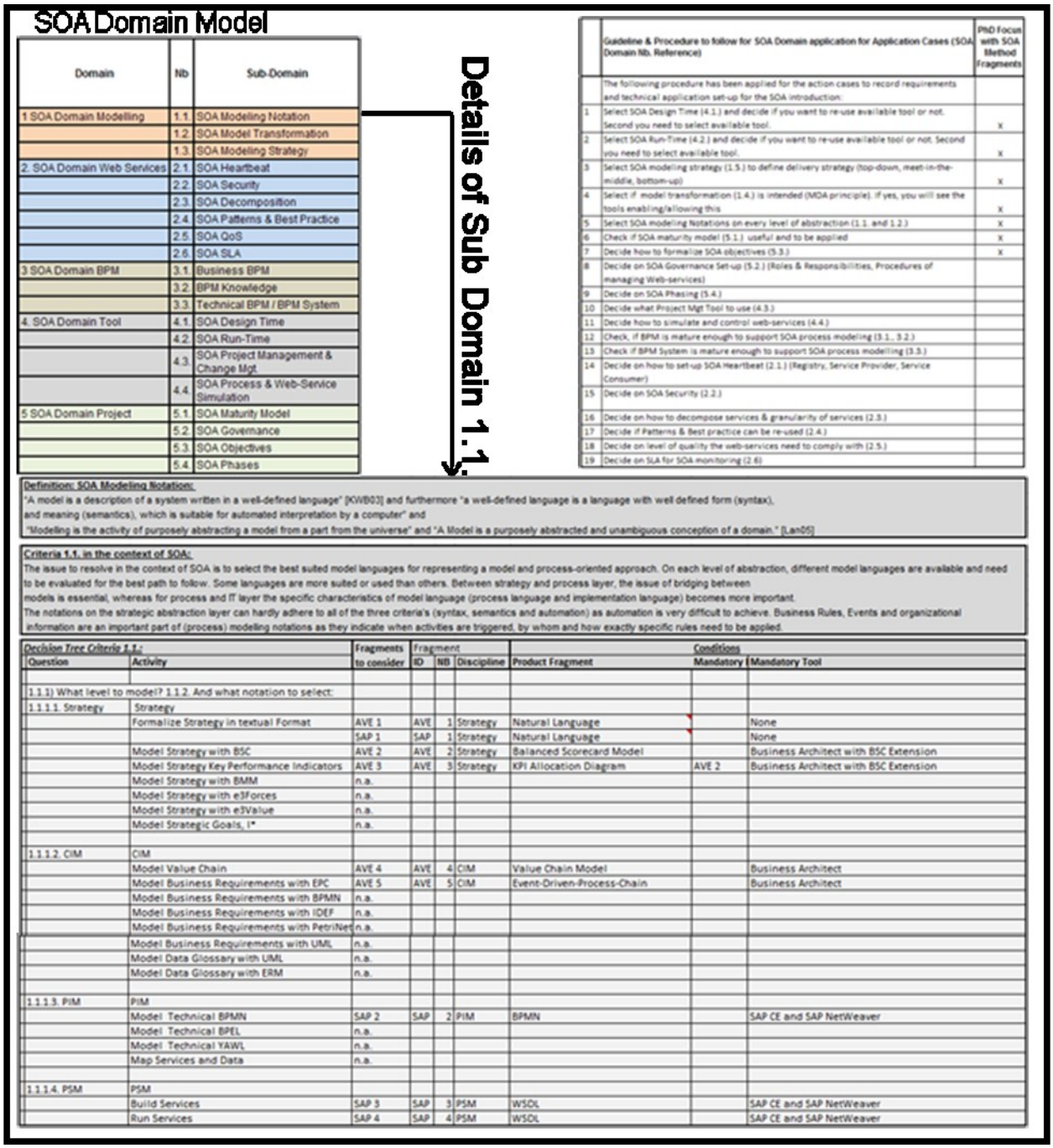
With these formalized set of method fragment in the EPF tool, the ground is prepared for the concrete application in two real SOA implementation field trials in chapter 6.

## 5.4. Facilitating Guideline Tool for SOA Domain Application

The objective of this tool is the support of the method fragment selection as described in figure 49. In order to select the matching fragments to the specific project requirements, required tools mandatory input/output and alternate solutions are evaluated.

For each SOA Domain Model sub-domain a definition and an explanation about the sub-domain context is given (chapter 3.1.). The excel tool recaps this information to allow the user to understand the sub-domain and to evaluate which activities (table 22) to select.

Ideally, for each sub-domain a separate worksheet should existing allowing guidance. Again, only the sub-domain "1.1. SOA Modelling Notations" is detailed. Within this sheet, the link to activities and fragment names (table 35) is done. The following example shown below illustrates the overview on the sub-domain 1.1. (SOA Domain Modeling):



SOA Domain Model

| Domain | Nb | Sub-Domain |
|---|---|---|
| 1 SOA Domain Modelling | 1.1. | SOA Modeling Notation |
| | 1.2. | SOA Model Transformation |
| | 1.3. | SOA Modeling Strategy |
| 2. SOA Domain Web Services | 2.1. | SOA Heartbeat |
| | 2.2. | SOA Security |
| | 2.3. | SOA Decomposition |
| | 2.4. | SOA Patterns & Best Practice |
| | 2.5. | SOA QoS |
| | 2.6. | SOA SLA |
| 3 SOA Domain BPM | 3.1. | Business BPM |
| | 3.2. | BPM Knowledge |
| | 3.3. | Technical BPM / BPM System |
| 4. SOA Domain Tool | 4.1. | SOA Design Time |
| | 4.2. | SOA Run-Time |
| | 4.3. | SOA Project Management & Change Mgt |
| | 4.4. | SOA Process & Web-Service Simulation |
| 5 SOA Domain Project | 5.1. | SOA Maturity Model |
| | 5.2. | SOA Governance |
| | 5.3. | SOA Objectives |
| | 5.4. | SOA Phases |

Details of Sub Domain 1.1.

| | Guideline & Procedure to follow for SOA Domain application for Application Cases (SOA Domain Nb. Reference) | PhD Focus with SOA Method Fragments |
|---|---|---|
| | The following procedure has been applied for the action cases to record requirements and technical application set-up for the SOA introduction: | |
| 1 | Select SOA Design Time (4.1.) and decide if you want to re-use available tool or not. Second you need to select available tool. | x |
| 2 | Select SOA Run-Time (4.2.) and decide if you want to re-use available tool or not. Second you need to select available tool. | x |
| 3 | Select SOA modeling strategy (1.5.) to define delivery strategy (top-down, meet-in-the-middle, bottom-up) | x |
| 4 | Select if model transformation (1.4.) is intended (MDA principle). If yes, you will see the tools enabling/allowing this | x |
| 5 | Select SOA modeling Notations on every level of abstraction (1.1. and 1.2.) | x |
| 6 | Check if SOA maturity model (5.1.) useful and to be applied | x |
| 7 | Decide how to formalize SOA objectives (5.3.) | x |
| 8 | Decide on SOA Governance Set-up (5.2.) (Roles & Responsibilities, Procedures of managing Web-services) | |
| 9 | Decide on SOA Phasing (5.4.) | |
| 10 | Decide what Project Mgt Tool to use (4.3.) | |
| 11 | Decide how to simulate and control web-services (4.4.) | |
| 12 | Check, if BPM is mature enough to support SOA process modeling (3.1., 3.2.) | |
| 13 | Check if BPM System is mature enough to support SOA process modelling (3.3.) | |
| 14 | Decide on how to set-up SOA Heartbeat (2.1.) (Registry, Service Provider, Service Consumer) | |
| 15 | Decide on SOA Security (2.2.) | |
| 16 | Decide on how to decompose services & granularity of services (2.3.) | |
| 17 | Decide if Patterns & Best practice can be re-used (2.4.) | |
| 18 | Decide on level of quality the web-services need to comply with (2.5.) | |
| 19 | Decide on SLA for SOA monitoring (2.6) | |

**Definition: SOA Modeling Notation:**

"A model is a description of a system written in a well-defined language" [KWB03] and furthermore "a well-defined language is a language with well defined form (syntax), and meaning (semantics), which is suitable for automated interpretation by a computer" and "Modelling is the activity of purposely abstracting a model from a part from the universe" and "A Model is a purposely abstracted and unambiguous conception of a domain." [Lan05]

**Criteria 1.1. in the context of SOA:**

The issue to resolve in the context of SOA is to select the best suited model languages for representing a model and process-oriented approach. On each level of abstraction, different model languages are available and need to be evaluated for the best path to follow. Some languages are more suited or used than others. Between strategy and process layer, the issue of bridging between models is essential, whereas for process and IT layer the specific characteristics of model language (process language and implementation language) becomes more important.
The notations on the strategic abstraction layer can hardly adhere to all of the three criteria's (syntax, semantics and automation) as automation is very difficult to achieve. Business Rules, Events and organizational information are an important part of (process) modeling notations as they indicate when activities are triggered, by whom and how exactly specific rules need to be applied.

| *Decision Tree Criteria 1.1.:* | | Fragments | Fragment | | | | Conditions | |
|---|---|---|---|---|---|---|---|---|
| Question | Activity | to consider | ID | NB | Discipline | Product Fragment | Mandatory | Mandatory Tool |
| 1.1.1) What level to model? 1.1.2. And what notation to select: | | | | | | | | |
| 1.1.1.1. Strategy | Strategy | | | | | | | |
| | Formalize Strategy in textual Format | AVE 1 | AVE | 1 | Strategy | Natural Language | | None |
| | | SAP 1 | SAP | 1 | Strategy | Natural Language | | None |
| | Model Strategy with BSC | AVE 2 | AVE | 2 | Strategy | Balanced Scorecard Model | | Business Architect with BSC Extension |
| | Model Strategy Key Performance Indicators | AVE 3 | AVE | 3 | Strategy | KPI Allocation Diagram | AVE 2 | Business Architect with BSC Extension |
| | Model Strategy with BMM | n.a. | | | | | | |
| | Model Strategy with e3Forces | n.a. | | | | | | |
| | Model Strategy with e3Value | n.a. | | | | | | |
| | Model Strategic Goals, i* | n.a. | | | | | | |
| | | | | | | | | |
| 1.1.1.2. CIM | CIM | | | | | | | |
| | Model Value Chain | AVE 4 | AVE | 4 | CIM | Value Chain Model | | Business Architect |
| | Model Business Requirements with EPC | AVE 5 | AVE | 5 | CIM | Event-Driven-Process-Chain | | Business Architect |
| | Model Business Requirements with BPMN | n.a. | | | | | | |
| | Model Business Requirements with IDEF | n.a. | | | | | | |
| | Model Business Requirements with PetriNet | n.a. | | | | | | |
| | Model Business Requirements with UML | n.a. | | | | | | |
| | Model Data Glossary with UML | n.a. | | | | | | |
| | Model Data Glossary with ERM | n.a. | | | | | | |
| | | | | | | | | |
| 1.1.1.3. PIM | PIM | | | | | | | |
| | Model Technical BPMN | SAP 2 | SAP | 2 | PIM | BPMN | | SAP CE and SAP NetWeaver |
| | Model Technical BPEL | n.a. | | | | | | |
| | Model Technical YAWL | n.a. | | | | | | |
| | Map Services and Data | n.a. | | | | | | |
| | | | | | | | | |
| 1.1.1.4. PSM | PSM | | | | | | | |
| | Build Services | SAP 3 | SAP | 3 | PSM | WSDL | | SAP CE and SAP NetWeaver |
| | Run Services | SAP 4 | SAP | 4 | PSM | WSDL | | SAP CE and SAP NetWeaver |

**Figure 55: SOA Domain Model for Situational Method application (Screenshot from Excel Tool)**

Concretely, an entry page with the complete SOA Domain Model and its sub-domains is provided. The figure above shows the details for the sub-domain 1.1. First, a definition of the sub-domain is given followed by the sub-domain context within SOA implementation project. Further down, questions are raised. In this case "What level of abstraction detail to model?" Second, "which notations to select on what level of detail?" On every question, activities can be identified. These activities have been introduced earlier in table 22. Each activity has one or more fragments for consideration. For instance for the strategy level, 7 activities are available, with in total 4 different and available fragments. Two fragments are proposed using natural language, one of these is using Balanced Scorecard and the last one is a Key-Performance Indicator Diagram. The fragment ID with the number is unambiguously identifying the fragment. The condition of mandatory input is giving advice if fragments can

be used without restrictions or not. As example, AVE3 can only be used after AVE2. Lastly, the mandatory tool field is indicating the tool which is enabling the fragment. In this case AVE2 and AVE3 product fragments need to be created by using ARIS Business Architect with BSC extension (Mandatory Tool). Next, there is a guideline as support for the application of addressing the SOA Sub-domains. The sequence proposal is based on the experience gathered from the application cases.

This support file is created manually and also updated manually. The support file in this first iteration cycle is somehow "hardcoded". This file could also be improved and made more explicit once more iteration cycles are achieved. There is no automatic link to EPF method fragment tool. This might be an improvement and future work as detailed in section 7.3.

## 5.5. Summary on Tooling & Prototyping for SOA Engineering Method

The presented work has shown how the original research question of an engineering method for SOA has been addressed in the last chapters and has implemented the conceptual work with the prototyping of a tooling support.

For the demonstration of application, content from SOA Value Engineering of IDS Scheer and SAPs Enterprise SOA has been selected. Both SOA methods have been analysed in chapter 3 where the qualification of sub-domain coverage in the SOA Domain Modelling has been illustrated. The method content has been sliced into method fragments and each fragment has been categorized with layer of abstraction and related sub-domain from the SOA Domain Model.

Process fragments (figure 53) were defined and also linked to the product fragments (figure 54). Additionally, work-steps (refer to table 30 for details) have been added (as very time consuming only to some of the fragments) enabling the user to understand how to achieve the indicated work products. For efficiency reasons, these method fragments have been documented and stored in the EPF and are available for re-use.

This chapter illustrated the static part of the tooling detailing how the information has been implemented. It is not detailing functionalities of the tools neither the process of using the tools. For this, embedded help-functionality or freely available web-tutorials can be used and are out of scope here.

Some important challenges within this formalization appeared:

First, the semantics between SPEM and ME needed to be aligned and understood. Therefore, the alignment table has been created.

Second, the provided methods are in various formats and supporting materials and sometimes not detailed enough to understand the content. Then, the content needs to be sliced and categorized. Eventually borderlines on abstraction layers are not always sharp enough or decisions need to be taken when formalizing the fragment. This decision, if content is not very clear, need to be captured and explained for the method fragments. Doing this, the risk is high to produce too much explanation where the method user could get lost. It seems to be more important to align used terminology avoiding confusion. Therefore, guidelines can be a helpful support in better understanding fragments by showing concrete examples, templates, checklists etc.

Third, the population of such a tool with method fragments for further re-use needs some knowledge on method engineering, tooling and also on content to be able to formalize method fragments on the right level of granularity. For efficient population, a method fragment population tool might be useful. This might be a further future development also stated in the final conclusion in section 7.3.

Finally, the formalization and population of method fragments into EPF is time-consuming if all relevant details are filled to allow non-specialists to achieve effectiveness and efficiency in method application. This is also a central feed-back from project teams during the field trial in chapter 6.

The next chapter will apply this approach in a real SOA project environment in order to identify strengths, weaknesses and validation of the proposed configuration process for situational SOA implementation.

# CHAPTER 6

# VALIDATION OF RESEARCH CONTRIBUTION BY FIELD TRIAL CASES



Chapter 6 will apply and validate the Configuration Process for situational SOA Method in two field trial cases. First, (section 6.1.) field trials are prepared and introduced. Next, the field trial objectives with related research questions to clarify as well as the field trial method (section 6.2.) are explained. Next, the evaluation of Configuration Process for situational SOA Method is conducted (section 6.3.). After this, the content of the method fragments is explored (section 6.4.). The conclusion is about the validation discussion on configuration process application, satisfaction on generated outcome as well as conclusions on applied research method (section 6.5.).

## 6.1. Preparation of Qualitative Field trial

The two real-life field trials will demonstrate practical application of the configuration process for situational SOA Method and is seeking getting answers to posed research questions in chapter 1. This is detailed in section 6.2.1.

Next, the field trials will allow the practical application and instantiation of method fragments. The two field trials are "typical" organizations in Luxembourg. Both are very different, but typical for Luxembourgish companies. Cargolux is the national airfreight transportation company, whereas LBBW bank constitutes a subsidiary of a major bank in Germany. These subsidiaries exist from all mayor banks mainly from France, Belgium, Germany, Italy and Great Britain.

We will only describe the application of method fragments which are related to SOA Domain modeling only.

## 6.2. Field Trial Objectives and Method

### 6.2.1. Field Trial Objectives

The objective of this field trial is to validate the process of managing situational context of organization as described in detail in section 4.1. in figure 48. The research questions posed in section 1.2.1. are worthwhile to be remembered:

***Q3.: How can the configuration process for SOA situational Method support the decisions taken in practice by organizations?***

***Q6.: What about the quality of generated SOA Method and the achieved results out of SOA Method?***

> ***Q6.1.: Is the quality of generated SOA Engineering Method satisfactorily?***
> ***Q6.2.: Is the achieved result from SOA Engineering Method satisfactorily?***

Research question 3 will be answered through textual description of the field trial, where question 6 requires the definition of validation criteria's for the posed questions. The evaluation is consisting of 3 processes

1. Define Evaluation Criteria's for
    a. quality of generated SOA Engineering Method.
    b. the achieved result from SOA Engineering Method.

2. Perform Field Trial by applying the Configuration Process for SOA Situational Method in two real projects.

3. Evaluate Field Trial by
    a. Qualitative observations and description of generated SOA Engineering Method.
    b. Perform Feedback-workshop with project group related to achieved results with SOA Engineering Method.

### 6.2.2. Field Trial Method

Following to Wieringa [Wie10], the field trial is a validation method under a controlled context with realistic examples. The method designer himself is applying or using the service/product to be investigated. In our case, the controlled context is a specific project, where we have been invited to apply the configuration process for situational SOA.

For both field trials we apply the described process in section 4.1.:

Start: Define evaluation criteria

---

1. Manage situational context of organization for SOA project
2. Selection of method fragment & assembly of method fragments
3. Perform project
4. Update method fragments after project end

---

End: Draw conclusions

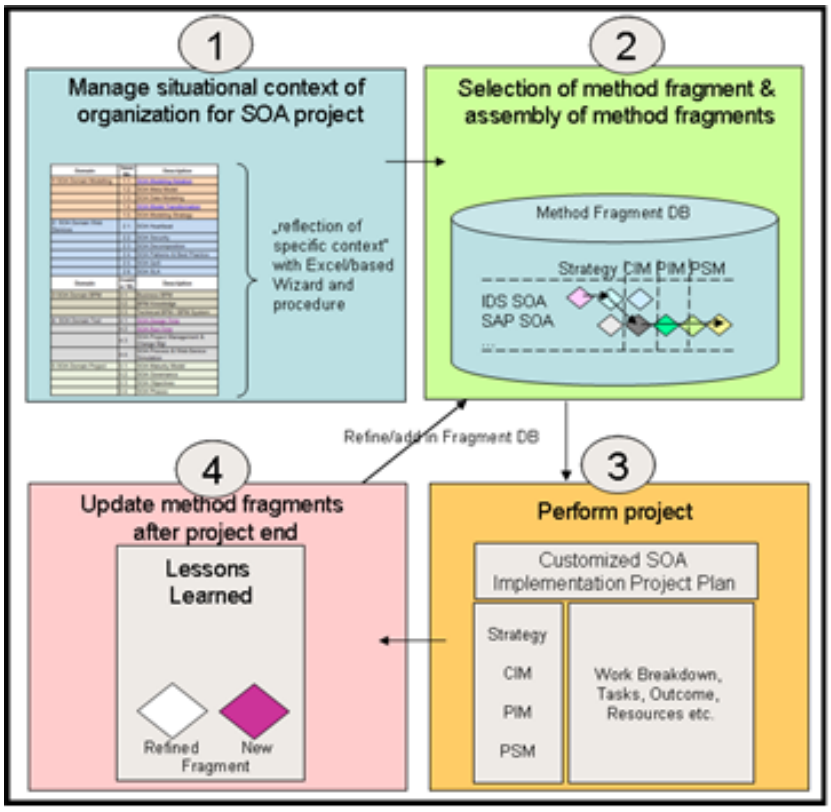



**Figure 56: Field trial Method for Method Fragment Application**

We are defining our own evaluation criteria's, which seems to be reasonable for the concrete context. The following table is summarizing these evaluation criteria's:

**Table 38: Evaluation Criteria's for Implementation Method Validation Q6**

| **Name of evaluation criteria** | **Evaluation Criteria Description for quality of generated configuration Process for situational SOA Method (Q6.1.)** |
|---|---|
| Usability | How easy is it to understand and customize SOA Engineering Method? |
| Repeatability | Can the SOA Engineering Method be re-used in another context or project? |
| Effectiveness | Is the method clear and complete in what to do and perform/apply the method? |
| Efficiency | Is the method efficient to perform and operate related to time, cost and quality? |
| **Name of evaluation criteria** | **Evaluation Criteria Description for result out of generated configuration Process for situational SOA Method (Q6.2.)** |
| Effectiveness of Method application result | Is the result usable for SOA Project implementation? |

These criteria's will identify if the applied method can be considered as successful or not.

At the end of the SOA project implementation, the CIO gave feed-back on the thesis evaluation objective. For the evaluation, a 4-range scale is used: 4: High Acceptance, 3: Acceptance, 2: Disagree 1: Fully Disagree.

## *6.3. Evaluation of Configuration Process for situational SOA*

### 6.3.1. Introducing Cargolux Airlines International SA

Cargolux, founded in 1970, is one of the leading cargo airlines worldwide, operating scheduled and charter services on a network covering all continents. The company offers almost 40 years of experience and, measured in ton-kilometers flown, today ranks in 9th position worldwide. In Europe, Cargolux is the largest all-cargo airline. The Airline is an integrated transportation company, operating exclusively for freight forwarders. Cargolux is using a fleet of 16 B747-400 freighter aircraft and 20 trucking contractors to move valuable and time-sensitive commodities through the worldwide network, covering over 90 destinations with over 1530 employees world-wide. The staff is multinational originating from over 30 countries. The network links many of the world's most important production centers of industrial, automotive and consumer goods through our hub in Luxembourg. Cargolux is constantly adapting its network to changing market demand and trade flows. and is specialized in the transportation of outsize shipments, perishable goods and live animals [CAR09].

Cargolux is directly and indirectly responsible for around 5,000 jobs in a small economy dominated by the financial services industry. For instance, Cargolux is the principal customer of Luxair Cargo Center, which employs over 1,000 people. Luxair is also Luxembourg's flag carrier and the Luxembourg state is its largest shareholder. Cargolux uses mostly Luxembourgish trucking companies. The Government is seeking to diversify the economy away from the financial services industry and the development of the logistics/transportation

sector is a cornerstone of that policy. Cargolux is contributing per year directly with over 310mEUR to the Luxembourg economy [CAR09].

Cargolux has decided for the next-generation Boeing 747-8F and was together with Japanese carrier NCA, the launch customer of this new aircraft type. Cargolux has mid 2012 4 new generation aircraft in its fleet. This decision shows the clear ambition to operate innovative aircraft allowing increased efficiency: less fuel burn, whilst the freight capacity is increasing.

The total income increased steadily since 2005 (figures in US$ Millions) [CAR11]:

| As at December 31, 2011 | 2011 | 2010 | 2009 | 2008 |
|---|---|---|---|---|
| Total income in US$ '000 | 1,898,641 | 1,748,431 | 1,352,259 | 1,984,675 |
| Profit/ (loss) for the year in US$ '000 | (18,337) | 59,838 | (153,301) | (61,043) |
| Shareholders' equity in US$ '000 | 455,243 | 504,901 | 469,508 | 454,240 |
| Tonnes carried | 658,800 | 683,380 | 627,813 | 703,601 |
| Tonne-kilometers flown (millions) | 5,039 | 5,284 | 4,800 | 5,411 |
| Available tonne-kilometers (millions) | 7,114 | 7,210 | 6,954 | 7,664 |
| Number of employees worldwide | 1,564 | 1,477 | 1,482 | 1,530 |
| Headquarters | 1,187 | 1,120 | 1,110 | 1,155 |
| Fleet | | | | |
| Boeing 747-400 Freighter | 10 | 13 | 14 | 16 |
| Boeing 747-400BCF | 3 | 1 | / | / |
| Boeing 747-8 Freighter | 2 | / | / | / |
| Aircraft on order | | | | |
| Boeing 747-8 Freighter | 11 | 13 | 13 | 13 |
| Aircraft subject to forward sale agreements | 0 | 3 | 4 | 6 |

**Figure 57: Cargolux Airlines Int SA Total Income 2008-2011**

The net revenue however decreased 2007, 2008 and 2009 three years in a row into a loss. 2010, Cargolux ended with profit of 60mEUR but due to striking crisis in 2011 again reported a loss end of 2011 [CAR11].

## 6.3.2. General Context of Cargolux

### 6.3.2.1. Strengths of Cargolux Airlines Int. SA

The fleet average is 8.5 years, which is young in relation to other carriers. The uniform B747 fleet allows low maintenance cost and efficient crew training. The specific type of Aircraft is low in fuel consumption, long range and high payload allowing fast turnarounds. This will even be increased through the new generation B747/8 Aircraft. In comparison to its predecessor, the new 747-8 features improved performance in terms of payload, range, environmental compliance through carbon emission (less 17%), noise reduction (less 30%) and fuel efficiency (less 17%) with modern “GEnx” engines from General Electric. It is 5.6

meters longer than the 747-400 and offers a payload capacity of 140 tons compared to 120 tons of the former model [CAR09].

Flexibility is one of Cargolux's strongest assets and the company successfully builds on long-term cooperation with their customers. In February 2000, Cargolux took into operation the world's first simulator for the B747-400 freighter. Other carriers e.g. Lufthansa send their pilots to Luxembourg for simulator training. Due to the flexible strategy of the network and good customer relationship, the crisis started to affect airfreight markets in June 2008, Cargolux has gained market shares in important markets like Germany, Italy and across Asia - always to the detriment of the home carrier. Except for its home base in Luxembourg, Cargolux does not have any significant ground infrastructure [CAR09].

Therefore, Cargolux can quickly open or close stations to respond to market conditions. While Cargolux will always be careful to avoid that its operational decisions harm its customers, this flexibility is a key advantage, in particular in a bear market.

Furthermore, the turn-over of the company is very low which is materializing in very experienced and knowledgeable employees with high seniority.

#### 6.3.2.2. Weaknesses of Cargolux Airlines Int. SA.

The organizational structure is mainly grown in history and very divisional or silo oriented.

Due to high seniority of employees, the dynamics related to change mgt. could sometimes be improved.

Only as of 2007, a professional governance structures for IT projects including a robust project management method has been implemented. Before that time, some bad experiences by the employees related to IT projects have been done.

The outsourcing fashion has been followed by preparing a spin-off in 2004 with the objective to outsource the complete IT department. This challenge has been done and a new IT service company called Champ Cargo Systems (CCS) has been created. The installed SLA between both companies was through the first years difficult to manage as there was no experience in that area. However, the situation is step by step improving. During 2007, Cargolux sold 51% to SITA and is holding still 49% of CCS shares.

Furthermore, two mayor threats consisting in exploding fuel price and the financial crisis have negatively impacted the results in 2007, 2008 and 2009 and gave Cargolux a very hard time. Consequently, IT budgets were downsized to a minimum.

#### 6.3.2.3. Preparing the ground for SOA at Cargolux

As mentioned before, a key fact to consider is the full IT outsourcing meaning that Cargolux has only 3 employees to manage the SLA, Cargolux IT strategy, IT projects and Business Process Management. CCS is serving its main customer Cargolux but also other customer airlines or handling agents. Therefore, both companies have similar but also divergent interests.

2005, Cargolux started a serious BPM project with the help of IDS Scheer to design and record processes in a structured way. This project took one year to finally reorganize processes with improvement potential bigger than 1mEUR, which was 5 time the cost for the project. Cargolux decided then to create a centre of excellence for BPM within IT to allow continuous improvement and a sustainable BPM effort. The top management sponsoring was crucial and the CEO involvement particularly motivating for the project team. The head of controlling and head of IT were driving the project forward to make sure a benefit could be materialized at the end of the day.

Consequently, BPM became a real asset to be re-used for all IT projects. In that period, a continuous improvement program, BPM Governance, BPM change Management were defined and implemented.

A massive investment into Master Data Management was done. All data has been modeled in ARIS. Cargolux was able to classify data into master or transactional data, who is responsible for it, who has what rights (r/w/m/d) and what application was hosting which data.

Next, EA has been built to show the links and relationship between the components e.g. Organization, Data, Applications, Processes, Strategy, Network etc. as well as guidelines and principles to follow.

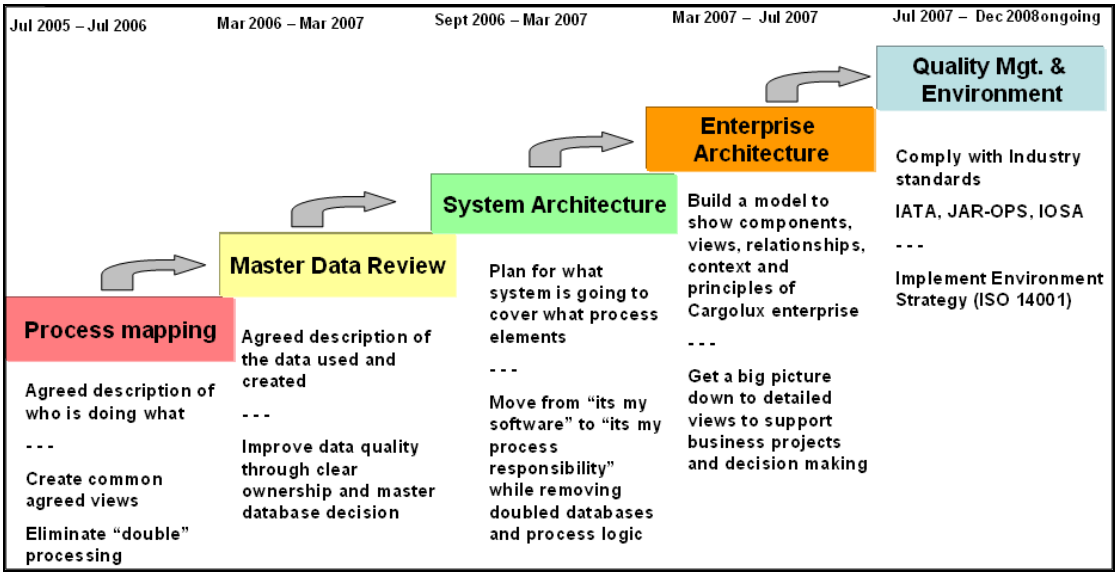



**Figure 58: Cargolux BPM Roadmap**

One of these principles is the SOA paradigm, which is materialized in the EA overview

By having these prerequisites for a successful SOA implementation, Cargolux was able to go forward with the concept, because considerable process assets/knowledge and BPM design tool were available.

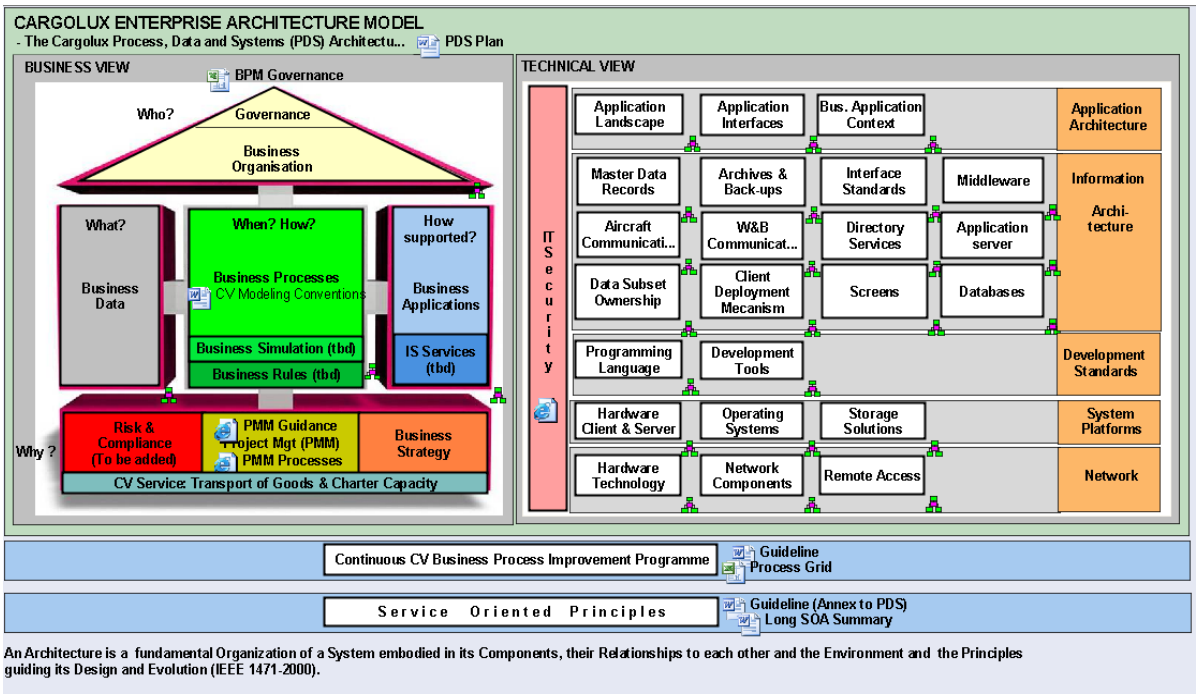



**Figure 59: Cargolux EA Model**

Overall, Cargolux is running 83 business applications and 15 desktop applications. The Cargolux processes are supported by the following main business applications support:

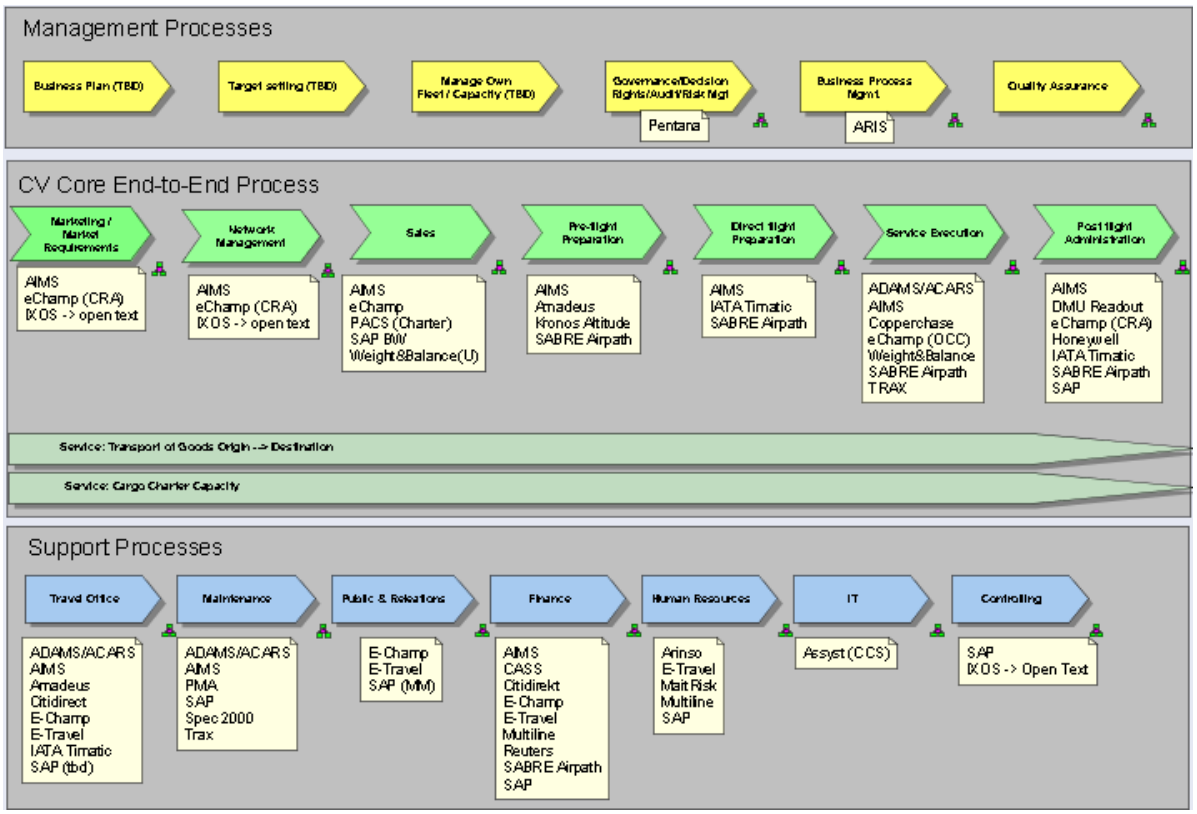



**Figure 60: Application support to main processes**

It is important to mention that for instance the application SAP can be divided into 18 sub-modules e.g. MM, SD, FI/CO, Treasury, BI, AM, OM etc.

AIMS for instance can be divided into 5 modules e.g. Training/Planning, Scheduling, Crew Assignment, Operations Control, and Tracking.

Cargolux BPM management knows exactly to the deepest level of detail what activities are supported and who is using the application.

For the middleware, Cargolux is using SAP Netweaver and SAP Process Integrator (PI), which will play an important role for the SOA IT architecture.

### 6.3.3. Manage situational context of organization for SOA project for Cargolux SOA Project

This section describes the application of the process "Manage situational context of organization for SOA project" as described in figure 48 (section 4.1.3.3.) to the Cargolux SOA project.

The process is starting with two activities on reading and understanding SOA Domains and sub-domains. This has been done based on information provided in the Method Fragment Wizard Tool. In our concrete case, this was the facilitating guideline tool (section 5.3.).

The third activity is about the decision on organization-specific priorities. This activity has been described through section 6.3.2., which permitted to harvest information on the identification of requirements and technology choices. Cargolux has decided to first start a proof-of-concept project limiting risks and investments. Furthermore, it has been decided to re-use available tools and technologies to avoid additional investments.

With the help of the facilitating guideline, activities were selected. It was checked, if and how the sub-domain "Modeling" was appropriately covered or not. If not, the risk of non-addressing was discussed.

Next, available method fragments were investigated and evaluated. The same has been done for available patterns. For this evaluation of method fragments, the details have been investigated in the method fragment tool/repository.

Next, there was no need to create a new fragment or go for back loop.

The following summarizing table is giving an overview of Cargolux situational context and constraints:

**Table 39: Summary of Cargolux Situational Context and Constraints**

| SOA Domain | SOA Sub-Domain | Cargolux Details |
|---|---|---|
| SOA BPM | Business BPM | BPM considered as strategic tool and philosophy used for various application cases (Documentation, Certification, Cost Reduction, Audit&Compliance, IT Developments, Enterprise Architecture etc.) by the entire company. |
| SOA BPM | BPM Knowledge | BPM Excellence Center with about 30 different modelers |

| | | |
|---|---|---|
| | | through divisions and BPM specialist as coordinator. Knowledge is centralized and recorded into system, procedure, processes and checklists. |
| SOA BPM | BPM System | BPM Modeling Tool available and in use since 2005, Process execution tool mainly SAP with its middleware SAP Netweaver and Process Integration. |
| SOA Tool | SOA Design Time | Re-use of available Modeling Tool ARIS, Purchase of SOA Module allowing generation of BPEL Model (but not used) |
| SOA Tool | SOA Run-Time | Re-use of available run-time technology SAP |
| SOA Tool | SOA Project Management & Change Mgt. | Use of SOA Domain Model and Method ME as proposed in this thesis. Project Management in EPF and MS Project. Change Mgt. Logs on processes in ARIS. |
| SOA Tool | SOA Process & Web-Service Simulation | Process and web-service simulation with SAP technology. |
| SOA Project | SOA Objectives | Risk-and cost limiting approach, therefore Proof-Of-Concept in terms of scope. (Travel Expense Mgt.)<br><br>A key success factor of the company is the flexibility in reacting to customer demand. This flexibility as a requirement and part of the business model, can be supported by SOA<br><br>Cargolux business objectives are oriented towards innovation to increase efficiency. This mindset is preparing the ground for investing into SOA.<br><br>Critical Success Factors for the SOA Project identified as well as link of business strategy to processes and IT.<br><br>SOA defined as a strategic objective for IT. |

### 6.3.4. Selection of available method fragments in Fragment Database

This section describes the application of the process ”Selection of Method Fragment” as described in figure 49 to the Cargolux SOA project.

The process is starting with the creation of a delivery process in the method fragment tool. This is necessary to ensure an own “process application” instance in the method fragment tool.

Next, delivery process attributes are filled, which are describing the project scope.

Based on first evaluation of method fragment candidate in previous process, method fragment attributes are checked clear and understandable. Based on this, Cargolux decided to select 6 fragments. The detailed selection description is summarized in table 40. An iteration of fragments was not necessary.

Next, the selected method fragments were manually compiled (by drag and drop) into the Cargolux delivery process by defining sequences. The selected method fragments are then consequently assembled into a coherent sequence. The work-breakdown structure for Cargolux, which is the Eclipse wording for an activity plan with selected fragments, can be shown as following:

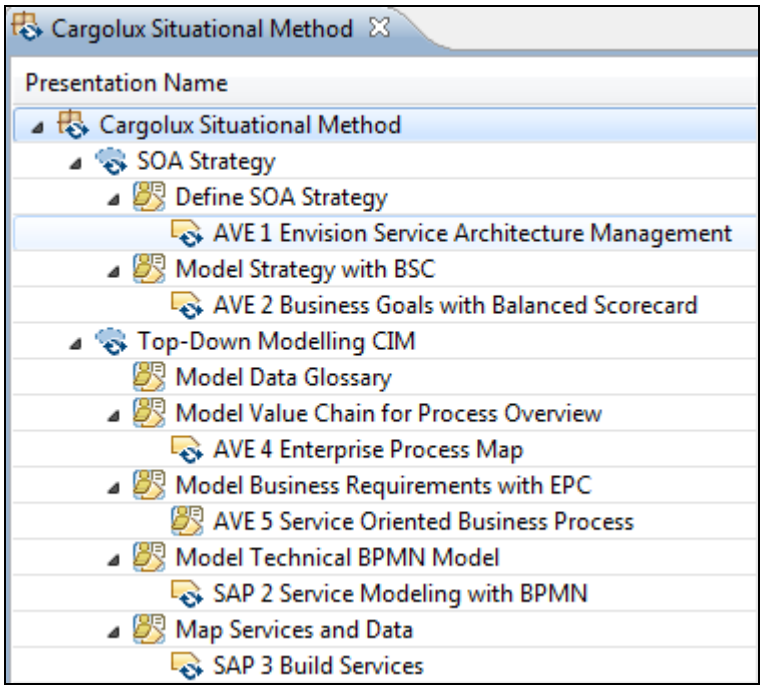



**Figure 61: Situational SOA Implementation Method Cargolux Action Case (Screenshot Eclipse)**

Each abstraction level (e.g. SOA Strategy) is decomposed into one or more activities (Define SOA Strategy), which is containing different tasks to be chosen. In this case, Cargolux has selected AVE1 Envision Service Architecture Management.

The selection of method fragments and putting activities into sequences is as stated earlier a completely manual task. This represents also an improvement direction, which is further discussed in section 7.3. future work.

A last control on SOA sub-domain coverage was performed. All sub-domains were addressed: The selected fragments were aligned and sequenced into a top-down approach (sub-domain 1.3.) addressing all layers of abstraction and product fragments as outcome (sub-domain 1.1.) Cargolux has decided not to use MDA principles (sub-domain 1.2.) as number of processes to transfer from design-time into run-time too limited and workload too high or not worthwhile.

As a last step, the chosen activities with related method fragments and sequence were transferred from Method fragment tool into MS project for proper project planning.

Based on the fragment list elaborated earlier, the following fragments have been selected:

**Table 40: Selection of method fragments for Cargolux Situational Method**

| Abstraction Layer | Activity | Method Fragment | Selection | Selection Description | Work Product |
|---|---|---|---|---|---|
| Strategy | Define SOA Strategy | AVE1 Envision Service Architecture | X | Textual Description of SOA objectives and benefits to be realized. | Natural Language |
| Strategy | Define SOA Strategy | SAP1 Discover Vision and Opportunity | | Not selected because AVE1 seemed to logically linked to AVE2. | Natural Language |
| Strategy | Model Strategy | AVE2 Business Goals with Balanced Scorecard | X | Balanced Scorecard Model used as work product to formalize strategy and objectives into a model BSC Model as method available in Modeling Tool | BSC Model |
| Strategy | Model Strategy | AVE3 Detail Key Performance Indicators | | Not selected as details on KPIs not necessary to formalize in a model as already textually stated. | KPI Allocation Diagram |
| CIM | Value Chain Modelling | AVE4 Enterprise Process Map | X | Value Added Chain Diagram (VACD) used as high-level diagram for process overview purpose | VACD Diagram |
| CIM | EPC Model | AVE5 Service Oriented Business Process | X | Event-Driven Process Chain (EPC) used as detailed process description in design-time environment. | EPC Model |
| PIM | BPMN Model | SAP2 Services Modeling with BPMN | X | BPMN used as technical model for process execution in SAP Process Integration engine. | BPMN Model |
| PIM | Web Service Identification | SAP3 Build Services | X | BPMN model enriched with Web-Services description | BPMN Model |
| CIM2PIM | | | | Reason for non-selection: 1.) Based on limited scope, transformation mechanism seemed to take more time than translating 5 processes manually from design-time tool ARIS into run-time tool SAP. 2.) As BPEL engine integrated into SAP middleware, no need to create BPEL model on PIM-Level in design tool. | |

Following to the method fragment selection, the solution path on abstraction levels is showing ARIS Tool for the design time on Strategy and CIM-level, whereas the technical implementation in SAP using BPMN and Web-Services for integration is on PIM and PSM level:

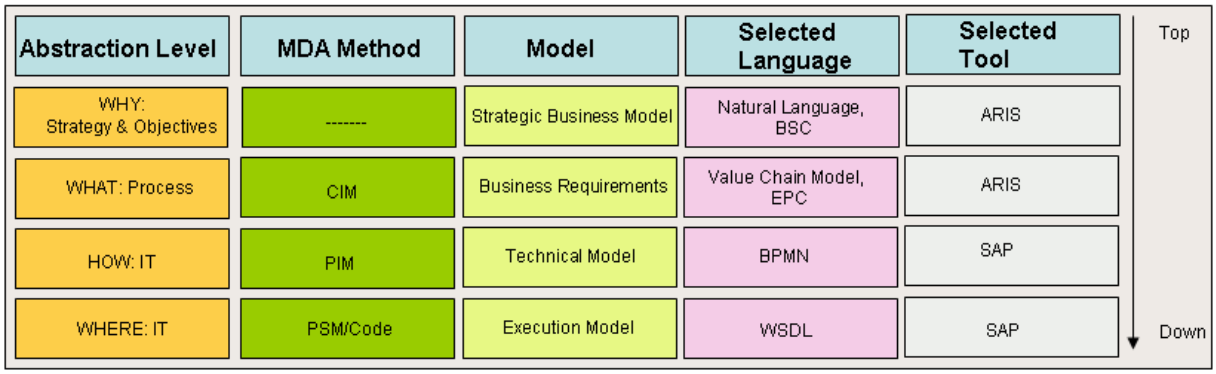



**Figure 62: Selected Modelling Languages for Cargolux Case**

Figure 62 summarizes the selected modelling languages/notations by Cargolux which is related to posed research questions 4 and 5. The strategy has been formalized in natural language (project brief, section 6.4.1.1.). However, strategic objectives were formalized with BSC (section 6.4.1.2.) model to allow executive committee to understand SOA objectives. These objectives were linked to high level CIM value chain model (section 6.4.1.3.). A drill down into detailed business requirement models (EPC) has been done (section 6.4.1.4.). The cut from SOA modelling into SOA execution has been done. For this, executable BPMN was used (section 6.4.1.5.) on PIM –level and has been enriched with web-service descriptions (WSDL, section 6.4.1.6.).

ARIS SOA Architect tool was used for the conceptual modeling, SAP for the technical modeling and the web-service run-time environment. Finally, it was rapidly evident that a full vendor-driven method could not be used, as a combination of two vendor tools with complete different methods and viewpoints had to be combined.

## 6.3.5. Validation Discussion on SOA Cargolux Application Case for Configuration Process for SOA Situational Method Cargolux Case

The validation discussion is decomposed into 2 key conclusions: First, is quality of generated SOA Engineering Method satisfactorily? Second, is the achieved result from SOA Engineering Method satisfactorily?

### 6.3.5.1. Feedback of Configuration Process for SOA situational Method Application Cargolux

This section is summarizing the application of both processes on the configuration process for situational method. The details of the Cargolux field trial fragments are further explained in section 6.4. As we have ourselves applied the method, this feedback is also our own observation.

The processes were easy to apply as the context and Cargolux situation was used to process-oriented approaches (section 6.3.2.) The processes were clearly defined, transparent and efficient. The alignment model was particularly important for the explanation of links between SOA Domain Model and Method Fragments (figure 42). The mix of method and tooling seemed to be balanced for this particular case.

Particularly interesting was the fact that the evaluation of method fragments was done by the project manager in discussion with the project team. This discussion fueled also the project team commitment, as the team had a common understanding and evaluation of selected method fragments. This discussion was possible by using the proposed SOA Domain Excel tool. Based on the detailed description of criteria's and the detailed description attributes of the method fragment (refer to chapter 4). Based on the SOA Domain Model criteria's, 4 AVE method fragments were combined and assembled with 2 SAP fragments. These fragments were selected in the EPF tool repository and assembled respecting conditions (e.g. AVE 1 is input for AVE 2).

In general, the SOA engineering method worked well, the CIO and the development team were satisfied. On usability, Cargolux was very positive, as the proposed Excel tool allowed rapid understanding for the SOA domains with the related questions to resolve. Despite the fact that the method could not be re-used in another project, they were confident that this would be possible to do. The CIO was satisfied with the way the method was described (SOA Domain Model, generic method processes, tooling) and the details provided on available method fragments. On efficiency, the update of fragments requires a lot of discipline after the project closure. Furthermore, it needs to be considered as an investment for next or upcoming implementations. As individual situations or context of specific fragments need to be detailed, this formalization was seen as time consuming. Practical considerations are sharply linked to permanent cost-benefit evaluations.

**Table 41: Feedback of SOA Engineering Method Application Cargolux**

| **Name of Feedback Criteria** | **Feedback Criteria Description** | **Scale:**<br>**4: High Acceptance/ Very good,**<br>**3: Acceptance/Good,**<br>**2: Not Satisfied/Disagree**<br>**1: Disappointed/Fully Disagree** |
|---|---|---|
| Usability | How easy is it to understand and customize SOA Engineering Method? | 4 |
| Repeatability | Can the SOA Engineering Method be re-used in another context or project? | 3 |
| Effectiveness | Is the method clear and complete in what to do and perform/apply the method? | 4 |
| Efficiency | Is the method efficient to perform and operate related to time, cost and quality? | 3 |
| | | |

Globally, the quality of generated method seemed to be very good. However, the complete method had to be applied manually. This could be dramatically improved to increase

efficiency and ease of use. Also the integration of the modelling tool, the method fragment tool and facilitating guideline tool could improve efficiency and increase user friendliness during application and generation of the configuration process. These observations are summarized in future work (section 7.3.).

#### 6.3.5.2. Feedback of Configuration Process for SOA situational Method Application Satisfaction Cargolux

This section details the feedback on the satisfaction with the result of the applied method. The Cargolux project team providing feed-back is including 5 people with different profiles (CIO Cargolux, Project Coordinator/Method Architect Cargolux, Project Manager Champ Cargo Systems, SAP Senior Consultant, SAP Senior Architect). The evaluation has been done in a workshop by explaining the feedback criteria description. All members gave their feedback on scaling followed by an alignment discussion on agreeing finally to 4 on the scale.

Globally, the result out of the applied SOA method was satisfactory. However, the two most important comments on weaknesses were seen in the limited number of available fragments and the manual selection and assembly process.

**Table 42: Feedback of SOA Engineering Method Application Result Cargolux**

| **Name of Feedback Criteria** | **Feedback Criteria Desription** | **Scale:**<br>**4: High Acceptance/ Very good,**<br>**3: Acceptance/Good,**<br>**2: Not Satisfied/Disagree**<br>**1: Disappointed/Fully Disagree** |
|---|---|---|
| Effectiveness of Method application result | Is the result usable for SOA Project implementation? | 4 |

This field trial showed clearly, that pre-configured and vendor-driven methods would have been difficult to apply, as they would not have respected the special conditions, available IT application landscape or scope to be applied. Consequently, a situation specific application seemed to be advantageous, as process fragments were applied depending on the situation or context. This is particularly true for medium size companies like Cargolux with limited IT budgets and the wish to re-use available tools and technology.

### 6.3.6. Introducing Landesbank Baden Württemberg

LBBW Luxemburg S.A. is a wholly owned subsidiary of Landesbank Baden-Württemberg (LBBW). LBBW Luxemburg has been operating in Luxembourg since 1978. Since 2007 with 270 employees and a balance sheet total amounted to € 14,080 mEUR, the bank is downsizing their activities. In 2011, the total balance sheet amount has been significantly reduced to 6,518 mEUR and the headcount decreased to 119 in 2011. LBBW Luxemburg also sold their Asset Management activities to Ycap Holding to react on crisis times with important downsizing

activities during 2010 and 2011. Since January 2011, the private banking activity has been sold and is now operated by Deka Bank Luxembourg [LBBW12]

During the field trial study with the AVALOQ-project, the bank was structured into 4 business lines:

Figure 63 is summarizing the different business lines [LBBW09b]:

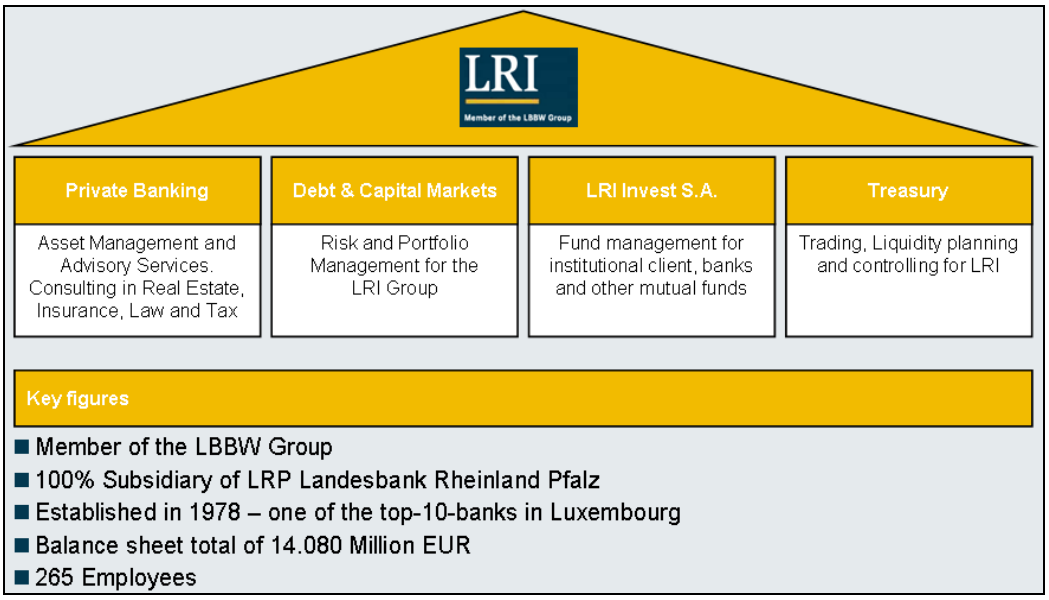



**Figure 63: LBBW Overview & Key Figures [LBBW09b]**

The LRI Invest has been sold to financial investors (Augur Financial Holding V.S.A.) which are running the fund management business to date.[LRI12]

## 6.3.7. General Context of LBBW

The case study projected started with LRI S.A. and was merged during the project into the LBBW S.A.. This was a first cultural change to digest with huge impact as organization structures changed and reporting lines to the new head office were established. Then, the financial crisis hit the market and the German Landesbanks suffered a lot as some of them were invested into sub primes and high risk investments. The losses of the German Landesbanks caused an important discussion within the German Government (as owners) about the business model of German Landesbanks. Despite the healthy and efficient Luxembourgish business model, the head office in Stuttgart decided to sell their major part of international branches to refocus more on credit and loan business for the German midsize market. With the confession of having failed in risk management and having revealed lacks in the business model and positioning, the LBBW was forced to act as prescribed by the government and the public opinion and decided to sell their Luxembourg branch within a transition period of two years meaning by end of 2011.

#### 6.3.7.1. Strengths of LBBW

The LBBW S.A. is a very profitable entity, despite the huge losses the mother company had with other international branches caused by high risk investments. The bank is very well organized and has a state-of-the art efficient process landscape with modern straight-through processes. The organization with the reporting lines is process oriented and the new core banking system can be considered as the latest state-of-the art system available on the market.

A high seniority and a very experienced employee's base is an asset of the bank.

#### 6.3.7.2. Weaknesses of LBBW

Firstly, the banking system "IBSY" was outdated and old fashion, a modern IT architectural style e.g. SOA not possible. This weakness has been recognized to increase business efficiency and realign business processes.

A weakness in this context is the permanent exposure and dependency to the mother bank, first situated in Mainz (LRI) and later after the merge in Stuttgart (LBBW). Important business lines have been already sold as part of a downsizing roadmap to digest the financial crisis.

Next, employees are not sure about the future owner and therefore incertitude could lead to the leaving of best staff.

#### 6.3.7.3. Preparing the ground for SOA at LBBW

SOA itself was not considered as an objective per se, but SOA came on the table as integrated part of a new core banking system. The old core banking system "IBSY" was out phased and simply not any more suited to serve the banking business in an accurate way. The bank had already since 2003 a comprehensive BPM system as part of the IT and Organization department. The bank is using ARIS Toolset and has a dedicated organization service of 4 FTE to care about processes and procedures. Through their BPM system, the bank had a very good knowledge on existing processes, what data and documents were used and how the IT systems were supporting the processes and activities. The organization knowledge also included methodological experience and the process management was well accepted and well known through business units. As said, SOA was an underlying mechanism and architectural style that enabled the new core banking system. To find the suitable system, the bank performed a detailed two year analysis project together with KPMG. A questionnaire with approximately 3000 questions has been developed and given to the software provider as a request for proposal (RFP). Within this extensive catalogue, software requirements and also underlying technological capabilities have been formalized. The project set-up can be summarized as follows [LBBW09b]:


**Figure 64: Project Set-up and Objectives LBBW [LBBW09b]**

As new system, the leading Swiss software provider for banking systems "AVALOQ" has been chosen and the consulting company "ORBIUM" as integrator has been selected. Once selected, a 6 months period prior to the project kick-off was used to develop the process-oriented implementation method. The implementation project itself took another 1.5 years for go-life. The summary of key figures and project organization structure [LBBW09b]:

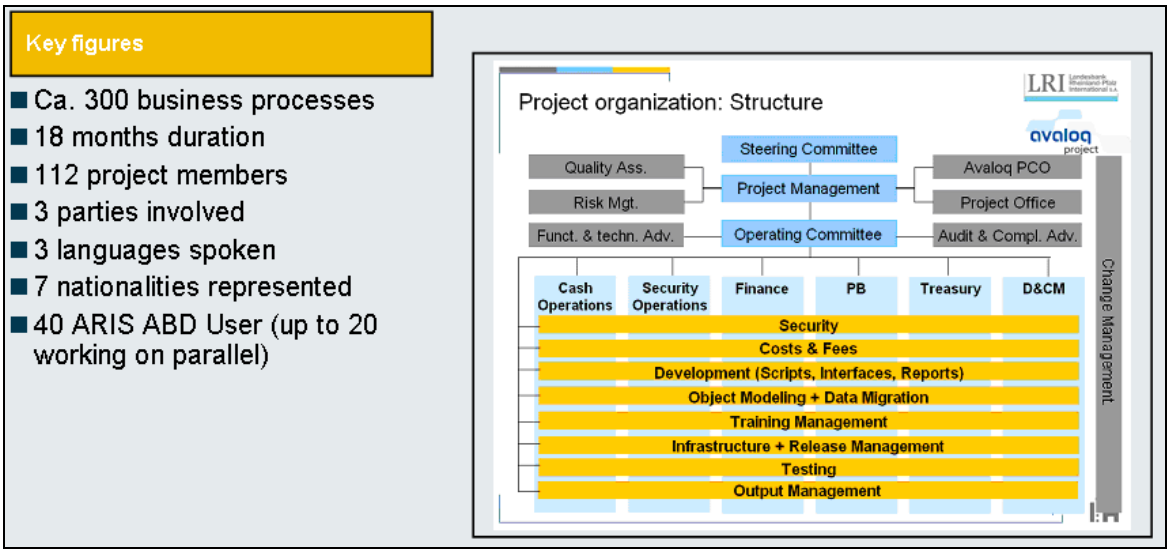



**Figure 65: Key Figures for Implementation Project**

## 6.3.8. Manage situational context of organization for SOA project for LBBW SOA Project

The section 6.3.7. explained the general context of LBBW bank. This section is related to the application of process as shown in figure 48 (section 4.1.3.3.).
To start the process, a kick-off meeting was held were SOA domains were presented and sub-domains further explained. Based on this, a workshop has been organized to gather all LBBW specific information as earlier presented in the kick-off. The most important situational context was described by a radical process and-organization re-engineering on the occasion of a new core banking system implementation. For the process design part, a modeling tool was on hand, whereas the banking core system AVALOQ has been selected.

With the help of the facilitating guideline, activities were selected. It was checked, if and how the sub-domain "Modeling" was appropriately covered or not. If not, the risk of non-addressing was discussed.

Next, available method fragments were investigated and evaluated. The same has been done for available patterns. For this evaluation of method fragments, the details have been investigated in the method fragment tool/repository. Due to the special set-up with AVALOQ as new core banking tool and related non-availability of AVALOQ SOA method, only AVE fragments have been evaluated.

Based on this, there was a need to create 2 new fragments: LBBW1 and LBBW2. The process "create method fragment" (figure 47) has been applied as the available AVE fragments were not covering the needs of the situational project context. The LBBW1 fragment has as objective to show details on used web-services per AVALOQ module. The generated product fragment is an access model (details section 6.4.2.4.) on PIM level. The LBBW2 fragment has as objective to show technical web-service process descriptions details. The product fragment outcome is under EPC notation. The conventions for the EPC notation are different than for AVE5. Consequently, an EPC product fragment is positioned on CIM level and another EPC product fragment on PIM level.

The situational context and constraints for the LBBW field trial can be summarized as follows:

**Table 43: Summary of LBBW Situational Context and Constraints**

| SOA Domain | SOA Sub-Domain | Context & Constraints |
|---|---|---|
| BPM | Business BPM | BPM is considered as strategic asset and used since 2001 by the internal organization department mainly for documentation, risk management, Activity-Based-Costing and IT developments. |
| BPM | BPM Knowledge | Organization department working since 2001 with process management. Knowledge is centralized and recorded into system, procedure, processes and checklists. |
| BPM | BPM System | BPM Modeling Tool available and in use since 2001, Process execution tool mainly Core Banking System IBSY to be replaced by AVALOQ. |
| Tool | SOA Design | Re-use of available Modeling Tool ARIS |

| | Time | |
|---|---|---|
| Tool | SOA Run-Time | Purchase of new Core Banking System AVALOQ |
| Tool | SOA Project Management & Change Mgt. | Use of SOA Domain Model and Method ME as proposed in this PhD. Project Management in EPF and MS Project. Change Mgt. Logs on processes in ARIS. |
| Tool | SOA Process & Web-Service Simulation | Process and web-service simulation with AVALOQ technology. |
| SOA Project | SOA Objectives | Full-Scope multi-million project, Replacement of outdated core banking system by new state-of-the-art technology using SOA principles to increase competitiveness by excellent lean processes.<br><br>Critical Success Factors for the AVALOQ Project identified as well as link of business strategy to processes and IT.<br><br>Single Business Processes to be offered as a service |

### 6.3.9. Selection of available method fragments in Fragment Database

This section describes the application of the process ”Selection of Method Fragment” as detailed in figure 49 to the LBBW SOA project. The process is starting with the creation of a delivery process in the method fragment tool for LBBW SOA project. Next, delivery process attributes were filled, which are describing the LBBW project scope. Based on first evaluation of method fragment candidate in previous process, method fragment attributes are checked clear and understandable. Based on this, LBBW decided to select 3 evaluated AVE fragments and to select 2 custom-made LBBW fragments. The detailed selection description is summarized in table 44. Next, the selected method fragments were manually compiled (by drag and drop) into the LBBW delivery process by manually defining sequences. The selected method fragments are then consequently assembled into a coherent sequence. The work-breakdown structure for LBBW, which is the Eclipse wording for an activity plan with selected fragments, can be shown as following:

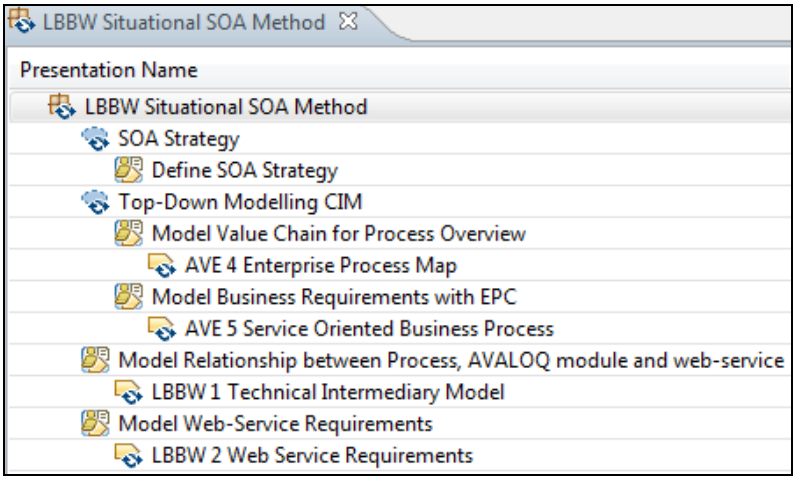



**Figure 66: LBBW Situational SOA Method Action Case**

The strategy was defined in AVE1 in natural language materializing in MS Word and MS PowerPoint documents. The formalization of the strategy in a strategy model was not intended as the added value seemed to be low. As overview models such as Org.Charts, Data Charts and IT Application Charts were already available, it was important to model value chain for scoping. In AVE4 the overview models were created. Here, two different levels of granularity were necessary. With AVE5, the central EPC model to explain business process requirements was modelled. The custom LBBW1 fragment serves as intermediary model to link process, AVALOQ module and web-service. The web-service details and behaviour are modelled with LBBW 2 changing a bit the EPC conventions. This choice is very interesting but understandable in the light that 1.) analysts know EPC very good, 2.) BPEL seems far too complex and technical, 3.) no technical transformation interface between design-time and run-time environments were available.

Based on the fragment list elaborated earlier, the following fragments have been selected:

**Table 44: Selection of Method Fragments for LBBW situational Method**

| Abstraction Layer | Activity | Task | Selection | Selection Description | Work Product |
|---|---|---|---|---|---|
| Strategy | Define SOA Strategy | AVE1 Envision Service Architecture | X | Textual Description of SOA objectives and benefits to be realized. | Natural Language |
| Strategy | Define SOA Strategy | SAP1 Discover Vision and Opportunities | | SAP not used | Natural Language |
| Strategy | Model Strategy | AVE2 Business Goals with Balanced Scorecard | | Value added of modelling strategy perceived as limited related to effort. | BSC Model |
| Strategy | Model Strategy | AVE3 Detail Key Performance Indicators | | SAP not used | KPI Allocation Diagram |
| CIM | Value Chain Modelling | AVE4 Enterprise Process Map | X | Value Added Chain Diagram (VACD) used as high-level diagram and for scoping | VACD Diagram |
| CIM | EPC Model | AVE5 Service Oriented Business Process | X | Event-Driven Process Chain (EPC) used as detailed process description in design-time environment. The web-service details, behavior, interfaces etc. were modeled again in EPC. | EPC Model |
| PSM | BPMN Model | SAP2 Services Modeling with BPMN | | SAP not used | BPMN Model |
| PIM | Model relationship between process, | LBBW1 Technical intermediary model | X | ***New fragment*** as intermediary model illustrating the relationship between process, AVALOQ | Access Diagram |

| | | | | | |
|---|---|---|---|---|---|
| | AVALOQ module and web-service | | | module and web-service necessary. | |
| PIM | Model web-service requirements | LBBW 2 Web-Service Requirements | X | ***New fragment*** as web-service requirements have to be detailed for web-service programming and implementation by AVALOQ programmers. | EPC Model |
| PSM | Web Service Identification | SAP3 Build Services | | SAP not used | BPMN Model |
| CIM2PIM | | | | None – as technically ARIS and AVALOQ were not able to interface and exchange/transform models | |

Following to the method fragment selection, the solution path on abstraction levels is showing ARIS Tool for the design time on Strategy, CIM-level and PIM-level, whereas the technical implementation in AVALOQ was done with web-services. Related to the project objective, the following modelling design path has been chosen:

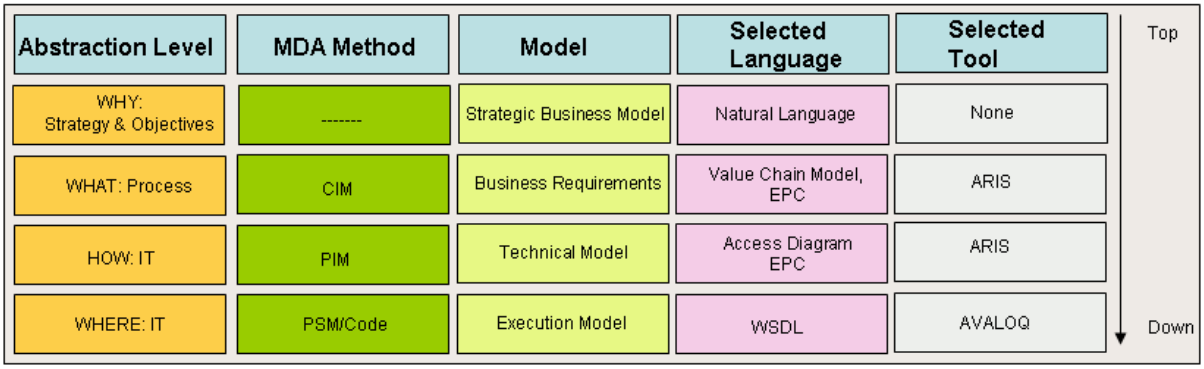



**Figure 67: Selected modelling languages for top-down modelling**

Figure 67 summarizes the selected modelling languages/notations by LBBW which is related to posed research questions 4 and 5. The strategy has been formalized in natural language ( section 6.4.2.1.) As a huge number of banking processes were addressed, it was necessary to use two different abstraction levels of value added chain models. A drill down into detailed business requirement models (EPC) has been done. Additionally, an access diagram has been used to bridge business requirements model and detailed web-service behaviour (EPC). More details can be found in section 6.4.2.1.

## 6.3.10. Validation Discussion on SOA LBBW Application Case for Configuration Process for SOA Situational Method Cargolux Case

Similarly to the Cargolux field trial study, the validation discussion is decomposed into 2 key conclusions: First, is quality of generated SOA Engineering Method satisfactorily? Second, is the the achieved result from SOA Engineering Method satisfactorily?

### 6.3.10.1. Feedback of Configuration Process for SOA situational Method Application LBBW

The main difference of generating the method was the fact that only 3 method fragments have been selected. The reason for this was the non-availability of SOA method from AVALOQ. Here, 2 new method fragments were created to mitigate the risk of missing formalization on PIM level. For this, the process as described in figure 48 has been applied successfully.

The non-availability of AVALOG method fragments was problematic, as the technical modeling and integration to the SOA design tool could not had been made. As AVALOQ method was limited, LBBW decided to create new LBBW fragments for PIM level.

The insight gained from this method application is the following: because auf situational context, a fragment has been re-used on two different levels, which was not foreseen at the beginning of the project. Therefore, the fragment AVE5 has been copied and modified to the PIM requirements of LBBW. The LBBW project created consequently its own fragment, which is a variant of AVE5. Instead of using it on the CIM-level for modeling the business requirements, the LBBW fragment has been used on the PIM-level for the specific purpose of detailing the web-service behavior/interaction and details. This was a consequence of non-availability of AVALOQ method. This example is demonstrating the successful application of the configuration process for situational SOA.

**Table 45: Feedback of SOA Engineering Method Application LBBW**

| **Name of Feedback Criteria** | **Feedback Criteria Description** | **Scale:**<br>**4: High Acceptance/ Very good,**<br>**3: Acceptance/Good,**<br>**2: Not Satisfied/Disagree**<br>**1: Disappointed/Fully Disagree** |
|---|---|---|
| Usability | How easy is it to understand and customize SOA Engineering Method? | 3 |
| Repeatability | Can the SOA Engineering Method be re-used in another context or project? | 3 |
| Effectiveness | Is the method clear and complete in what to do and perform/apply the method? | 4 |
| Efficiency | Is the method efficient to perform and operate related to time, cost and quality? | 3 |

The project manager mentioned that detailed knowledge on ME and SOA is necessary to apply the method and the related SOA Domain Model. Next, the workload to formalize fragments is considerable, but at the end of the day worthwhile doing it under the condition not to over-engineer the approach. Similar to the first case, the wish to have a smart wizard for fragment creation, selection and assembly would be useful to increase efficiency in method fragment creation and application.

Next, based on this field trial experience, AVALOQ announced the wish to create own method fragments to be used for coming projects to overcome the lack of method availability.

#### 6.3.10.2. Feedback of Configuration Process for SOA situational Method Application Satisfaction LBBW

The LBBW project team was approximately 23 persons from the total team of 112 project team members. As explained in figure 65, the project management and applied method was part of the project office. Therefore, the CIO being the responsible project manager and heading the project office gave feedback together with the key resources in the project. Similarly to the Cargolux case, a discussion took place to agree on effectiveness of method application result. The rating was estimated "acceptance/good":

**Table 46: Feedback of SOA Engineering Method Application LBBW Result**

| **Name of Feedback Criteria** | **Feedback Criteria Description** | **Scale:**<br>**4: High Acceptance/ Very good,**<br>**3: Acceptance/Good,**<br>**2: Not Satisfied/Disagree**<br>**1: Disappointed/Fully Disagree** |
|---|---|---|
| Effectiveness of Method application result | Is the result usable for SOA Project implementation? | 3 |

Generally, the project situation was complex, as 3 parties (LBBW, AVALOQ, Orbium) with different viewpoints and objectives were involved into the project. The required steps and activities were build-into the overall project plan. The project management and the method-team worked closely together, and communicated to project team groups about requirements.

Overall the generated situational method was not as efficient as expected, as the complete project was more seen as a "new core banking system" and not a "SOA project". SOA objectives were implicitly described by the project objectives. The centre of process excellence was managing the processes and method acting as project office and quality assurance body consisting of more people than in the first trial case. This fact made the communication and alignment effort with project teams much higher.

In total, the implementation of the new AVALOQ core banking system was successful meaning in time, in budget and in required quality. SOA principles have been successfully implemented by allowing web-services to perform business activities but also to interface business functionality such as data interfacing with other applications. As AVALOQ had no dedicated process-oriented SOA implementation approach, the complete conceptual design

part including technical models (behaviour of web-services) have been done in ARIS Business Architect.

## *6.4. Method Fragment Details of Trial Cases*

The following sections will focus on the Cargolux field trial and describe in detail the applied method fragments, starting with Strategy layer going downwards into Processes and IT layers. The content of selected method fragments as described in section 4.2. is now further explained to give more details on research questions 4 and 5 which are the question about candidate modeling languages and the integration of product fragment artifacts into the applied configuration process for situational SOA Method.

### 6.4.1. Cargolux Method Fragment Details

In order to show the application of the method, it is also important to illustrate the content to ensure proper understanding of the approach. The following subsections will detail and illustrate the produced content of selected method fragments for Cargolux field trial.

#### 6.4.1.1. Fragment AVE 1: Envision Service Architecture Management (Activity: Define SOA Strategy)

The chosen strategy was consisting in first defining a Proof-of-Concept (PoC) to allow a first experience on small scale instead of taking a high risk in uncertain times for a full blown project.

As Cargolux faces a fragmented and heterogeneous application landscape, the SOA paradigm seems to be promising. The following potential benefits have been identified:

**Reduced time to market (TTM) for new services**
Reduced TTM is achieved by enabling IT architects and developers to focus their efforts more on developing and delivering unique business service logic, and less on middleware. Furthermore reduced TTM is increased by defining standard business processes and associated infrastructure to allow choreography of services based on "business process models" (BPM).

**Reduced total cost of ownership (TCO) of IT infrastructure and business services**
Eliminating costly, proprietary middleware and replacing it with equally capable, open standards-based Web services technologies (SAP Process Integrator (PI) engine available for Cargolux). Second, another advantage persists in consolidating well-defined business functions into services that can be shared by multiple business units

**Enterprise Agility**
Cargolux is looking for the ability to quickly react to changing conditions by configuring and extending quickly application functionality. Second argument is the agile reconfiguration of technical infrastructure and organizational structure as business requirements change because it becomes easier to add, remove, or modify services than to change hard-corded applications.

**Securing IT Investments**
Another benefit identified by Cargolux consists in a better protection of IT investments on the long term due to service encapsulation. The interface of the service may change while protecting the internal code, or vice versa, the internal code can be upgraded without affecting the rest of the architecture.

**Business and IT Alignment**
Alignment of IT capabilities with business goals is made easier because of the modular and dynamic structure of SOA-based environments.

**Ensures Data Quality**
The service orientation unlocks data and functions from monolithic applications and favors reuse of functionality. SOA concept enforces clean data management practice.

**Promoting Standard Processes and Best Practice**
The service orientation promotes standardized services and the usage of best practice web-services (e.g. SAP repository).

**Enabling CV to integrate better common processes with customers and suppliers**
Web-services will allow a much better integration into customers and suppliers processes by providing functionality that is decoupled from IS Systems.

**Breaking Silo Mentality**
Web-service will provide business functionality depending on processes. The application used is not important anymore.

The Cargolux strategy set-up and the matching to the top-down pyramid can be shown as follows:

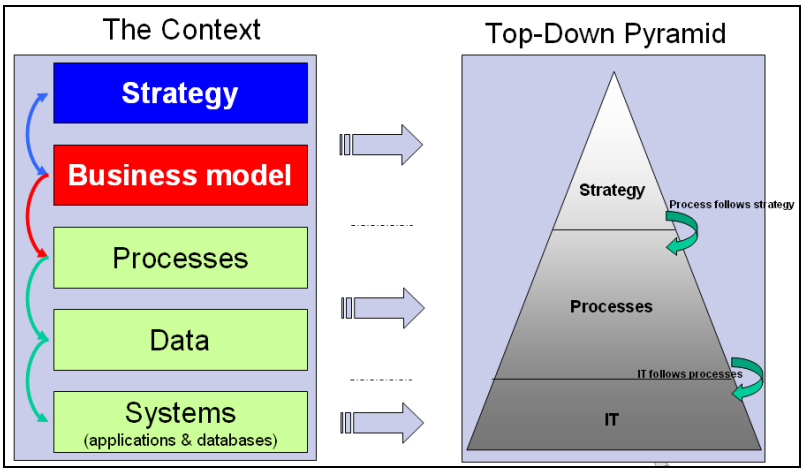



**Figure 68: Cargolux SOA Strategy Context**

Cargolux has a well-defined market strategy and an underlying business model, which is called "Cargolux heartbeat". The business model is relying on a very flexible network to serve in the best way customers. Customers are defined uniquely as forwarders e.g. Panalpina, Kuehne&Nagel or DHL. The green indicated areas process, data and systems is owned by the divisions and facilitated by the IT department.

Within the SOA Strategy, a detailed analysis on Strengths, Weaknesses, Opportunities and Threats (SWOT) has been conducted to obtain a better picture:

As major strengths, Cargolux identified the knowledge on BPM as critical success factor for SOA construction as well as theoretical SOA knowledge. Furthermore, the top management is a strong supporter of IT division with its vision for SOA. As SOA has practically not been implemented so far, it is evident that the missing experience has been identified as a potential weakness. The timing with 2009-2011 was clearly an advantage as the crisis allowed to concentrate on internal conceptual work. Traditionally, Cargolux is not a pioneer for the latest technology or fashion but follows very close by monitoring exactly if a new technology is worth to be adapted. The pitfalls of heavy investments at high risk are by this approach reduced to a minimum. As Champ Cargo Systems (CCS) which is the outsourced full IT provider of Cargolux has the latest technology available, it was evaluated also as opportunity to convince CCS to go towards a SOA evaluation exercise. This fact is also a major threat as CCS has so far no experience with SOA and therefore lacks on method, experienced staff. Cargolux is completely dependent on the ability to execute this type of projects, as the implementation of the conceptual work is completely done on CCS side.

The conclusion on this SWOT was to use SAP technology with CCS as Cargolux is dependent on their resources in medium and long term. SAP consultants were used as "train-the-trainers" for CCS SAP specialists.

#### 6.4.1.2. Fragment: AVE2 Business Goals with Balanced Scorecard (Activity: Model Strategy with BSC)

The selection for the SOA Proof-of-Concept has been done on an extensive process landscape analysis, which has been driven by strategic objectives of the company formalized in the Balanced Scorecard Model:

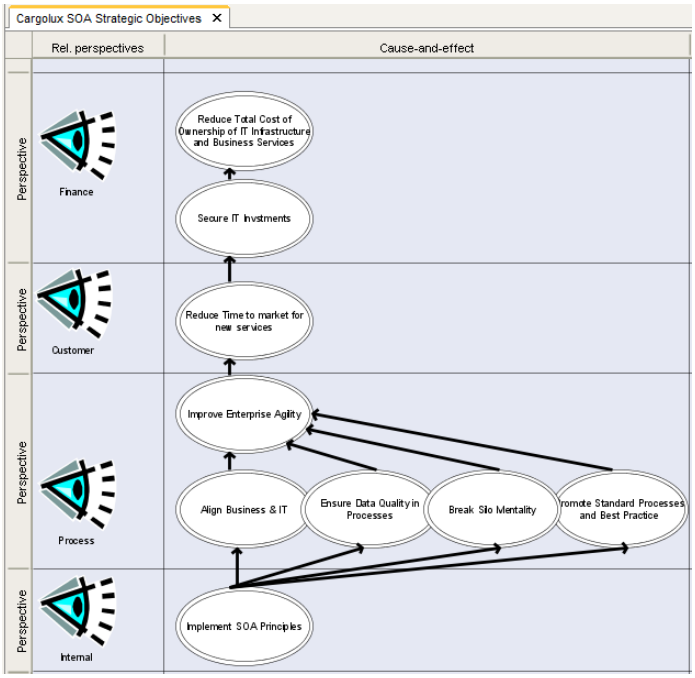



**Figure 69: Cargolux Balanced Scorecard Model with Cargolux SOA Objectives**

This BSC Model allowed formalized strategic objectives as defined in AVE1 method fragment in table 34. With this application of strategic modeling, the SOA initiative is embedded into a strategic landscape and method allowing a formalized way into the SOA subject. The outcome shows the strategic objectives positioned on the 4 BSC views as described in section 2.2.4. The relationships between objectives are "cause-and-effect" connectors. A possible weighting of "cause-and-effect" influence has not been done, as this was very difficult to argue.

Independently from the pure modelling part, the strategy also includes the SOA Governance definition with roles & responsibilities, policies and procedures. This is not further detailed as this is not related to the focus of showing the top-down model and process-oriented SOA implementation.

Furthermore, one process has been selected to improve dramatically performance, namely "Travel Expense Mgt." to prove rapidly the value of SOA principles. This is a high volume process with a considerable amount of interfaces to other systems to retrieve and re-use information. As the underlying technology is mainly SAP, this fact was another argument to stay within SAP run-time environment. The completely manual and paper oriented process should be transformed into a high-performance process using leading technology (SAP portal, SAP PI, SAP CE) to dramatically improve the process. Once feasibility of SOA and also benefits illustrated, the further roll-out was planned in a second phase. This is a good example how scope, culture and available IT Architecture influence SOA objectives.

#### 6.4.1.3. Fragment: AVE4 Enterprise Process Map (Activity: Model Value Chain for Process Overview)

In order to define the business requirements, it is necessary to model the process first from the business user perspective. In order to have a good overview of the processes in scope, a high-level diagram is used. Furthermore, the relationships and dependencies in between process are illustrated by using a value chain model:

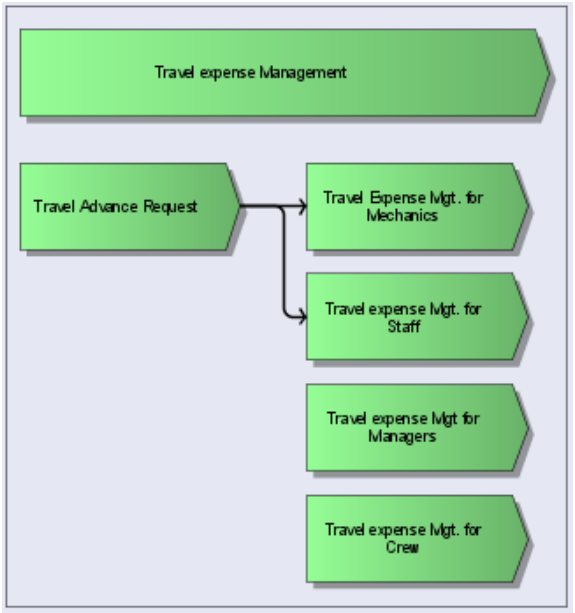



**Figure 70: Value Chain Model for scoping and high-level Process Overview**

Five process scenarios have been identified within the scope:

1.) Travel Request Advance,
2.) Travel Expense Management for Managers,
3.) Travel Expense Mgt. for Mechanics,
4.) Travel Expense Mgt. for Staff
5.) Travel Expense Mgt. for Crew

Different guidance documents have helped as input to identify the specificities such as the "Collective Work Agreement" and "Cargolux Travel Policy". Each process in the high-level process map has been detailed on activities level. Beside the standard EPC information with events, activities, positions and application systems, the process models were enhanced with required information such as data used, potential web-services used and potential screens. Each task has been drilled down into work steps and allowed therefore a very detailed explanation of business requirements and system behaviour. The result out of this business requirements analysis was a functional blueprint document as outcome.

#### 6.4.1.4. Fragment: AVE5 Service Oriented Business Process (Activity: Model Business Requirements with EPC)

As described in section 6.3.4., Cargolux decided for EPC as modelling language for representing business requirements. All other available Cargolux processes (approx. 700) were since 2005 designed with EPC-method and therefore well-known by analysts and users. EPC notation is allowing business analysts to formalize the requirements which are then understandable for IT roles. Each of the activities is further described on work-step level to unambiguously make requirements clear how the user and the system should work. Figure 71 illustrates the process for managers "Travel Expense Management for Managers":

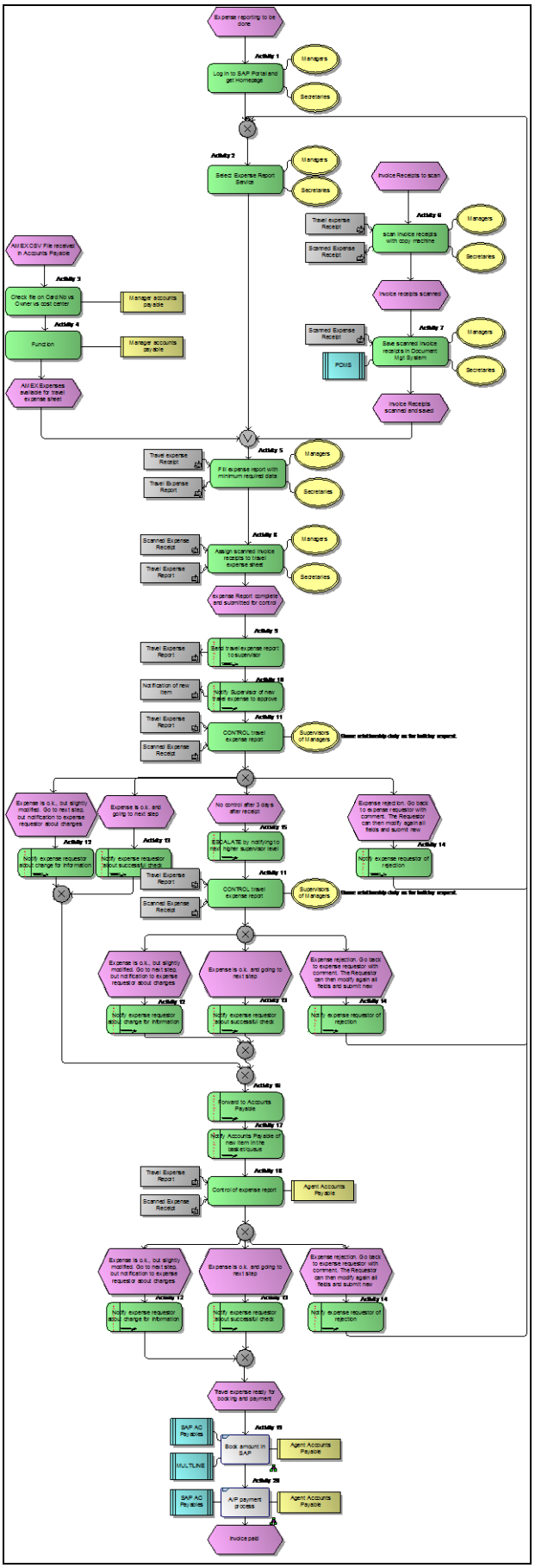



**Figure 71: Example for Cargolux Functional Requirement Process, EPC, CIM Level**

As the processes need to be implemented and executed in an operating environment, it is necessary to think about the interface between the "design-time" and the "run-time". As SAP has been selected as operating environment, the next task was to transfer in the most efficient way the EPC model into an executable BPMN model in SAP configuration environment. This was done manually, as the exportation in XML format and the re-work in SAP BPMN would have taken longer time than doing the transformation manually from design-time into run-time environment.

#### 6.4.1.5. Fragment: SAP2 Services Modelling (Activity: Model Technical BPMN)

SAPs "Service Modelling" fragment is modelling and designing the BPMN process containing the identification of SAP Business Objects, the description of user interfaces and interaction and the determination of Service Candidates. The EPC-Model containing the business requirements is translated manually into BPMN and enriched with required technical information.

The modelling notation in SAP is BPMN 2.0., which is executable in the SAP SOA Run-time environment. As an example illustrating the difference between these levels of abstraction, the technical process is shown following BPMN 2.0. Modelling conventions:

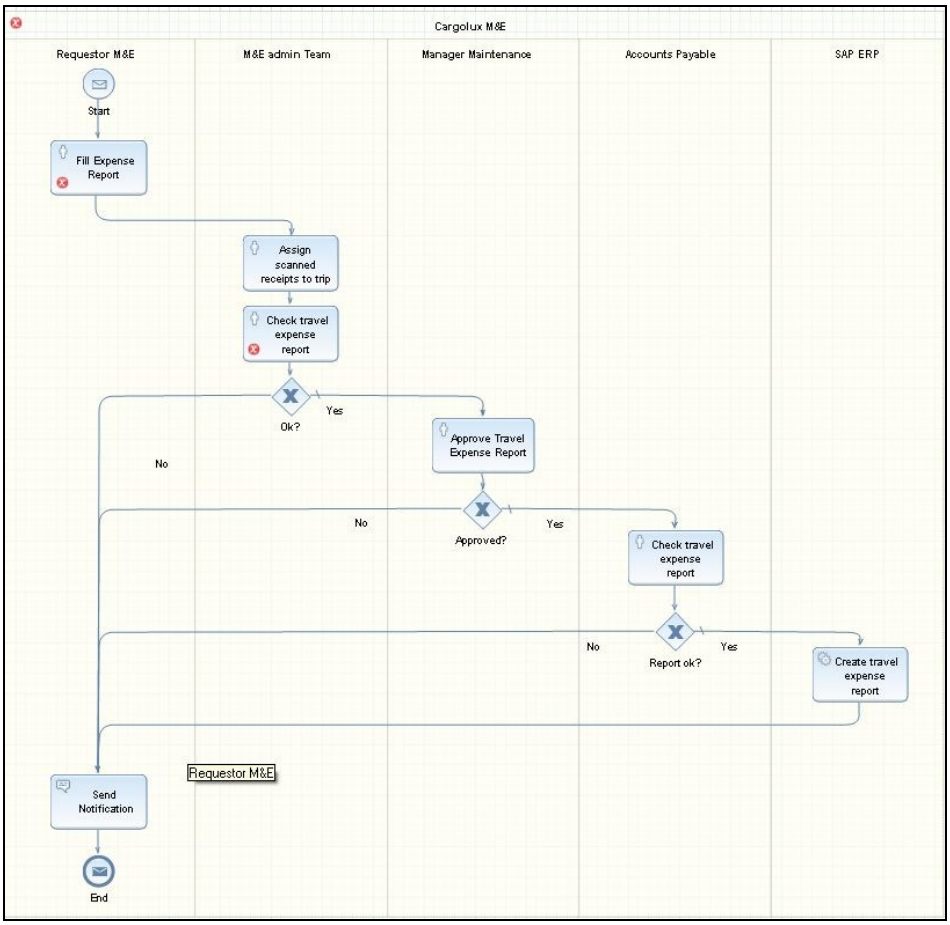



**Figure 72: Example for Cargolux Technical Process, BPMN, PIM Level**

The actors are positioned at the top of the swim lanes, which are executing activities. Decision flows and rules are used.

#### 6.4.1.6. Fragment: SAP 3 Build Services (Activity: Create Services with WSDL)

Every activity in the technical model gets a screen assigned, which is built in Web DynPro. Related to this, services are re-used from the Enterprise Services Repository (ESR) which is an integral part of PI and SAP CE components. It is used to design, model, manage and discover enterprise SOA based objects. The following table illustrates (an excerpt) of the list with required web-services derived from the business requirements. Some of these are already available as best-practice in the SAP Registry. Others are already available and can be re-used because of earlier web-service developments. 60% of total number of required services need to be developed.

**Table 47: Excerpt of Required Web-Services List**

| ***Name*** | ***Description*** | ***Web-Service available in Registry*** |
|---|---|---|
| *...* | *...* | *...* |
| *BAPI_TRIP_CREATE_FROM_DATA* | *Create trip in SAP Travel Management with trip header data and expense items.* | *Yes* |
| *ZFI_TRIP_ADD_ATTACHMENT* | *Add PDF attachment to trip created in SAP Travel Management.* | *No* |
| *ZFI_TRIP_APPROVE* | *Approve trip and change status to 'settled'.* | *Yes* |
| *ZFI_TRIP_POST* | *Post trip in FI Accounting and trigger payment to employee.* | *No* |
| *ZFI_TRIP_GET_OWN_LIST* | *Get list of all trips for a given employee ID.* | *No* |
| *ZFI_TRIP_GET_APPROVER_LIST* | *Get list of all trips for a specific approver.* | *No* |
| *BAPI_TRIP_GET_DETAILS* | *Get trip details for a given employee ID and trip number.* | *Yes* |
| *ZFI_TRIP_GET_ATTACHMENT* | *Get PDF attachment of a trip to display in CE portal.* | *No* |

Again, as the focus of this present thesis is lying on the process- and model-driven part, the technical SOA Web-Service issues are not detailed but only enumerated in table 47.

### 6.4.2. LBBW Method Fragment Details

The following subsections will detail and illustrate the produced content of selected method fragments for LBBW field trial.

#### 6.4.2.1. Fragment AVE 1: Envision Service Architecture Management (Activity: Define SOA Strategy)

LBBW S.A. was considering SOA not being the first strategic objective by itself, but acting as an enabler for the new core banking system with the main goal to increase efficiency and therefore to significantly reduce cost. Interestingly, this primary objective gave the opportunity to radically re-engineer the whole organization and to structure the business units with corresponding organizational impacts strictly following products and end-to-end processes. Therefore, the second main objective was a consequent alignment of the overall banking organization with the products and services and realize efficient straight through processing. The long term objective (5 years) is to offer single business processes as a service. IT is playing the role of service provider towards internal needs. It is also planned to analyse possibilities to sell these services to the external e.g. other banks. To enable the technical platform, service orientation has been introduced as basic principle for the IT architecture.

The new system brought in nearly 6000 functionalities or services which were structured into 6 groups:

- Cash Management,
- Securities,
- Deposit,
- Investment Funds,
- Administration: Master Data and
- Administration: Private Banking Sales Support.

The organization strategy with the mentioned objectives, were input for the project preparation as well as the organization culture and the IT Budget with two-digit Mio Euro. Unfortunately, the Return on investment has not been calculated the business case was valid and top management support obtained. Roles and Responsibilities were agreed. The project office, quality assurance and risk management were assured by the organization department also responsible for the BPM system.

Similar to the first case study the following principles and decisions have been taken:

- Top-down approach,
- Knowledge of Processes and Process Documentation,
- Tool driven approach,
- Extensive Change Management,
- Holistic Approach

The strategic IT roadmap towards long-term objective SOA [LBBW09b]:

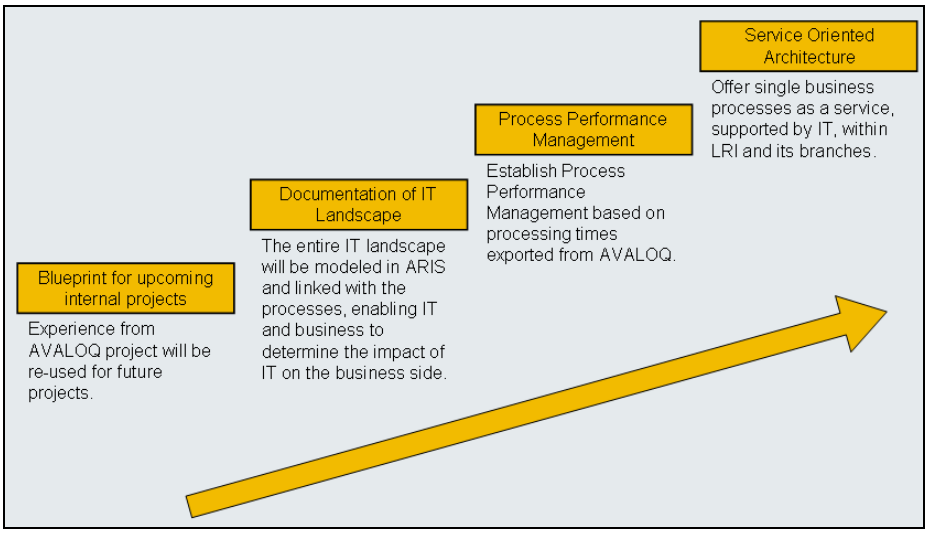



**Figure 73: Strategy for AVALOQ Project [LBBW09b]**

The box on the upper-right corner in figure 73 indicates clearly SOA as a strategic objective. The modeling of strategy has not been done, as the description in natural language and presentations were satisfactory for the project group.

### 6.4.2.2. Fragment: AVE 4 Enterprise Process Map (Activity: Model Value Chain for Process Overview)

Figure 74 illustrates a high-level strategic business process landscape showing the executed processes by the bank. Due to complexity, it is necessary to model details of illustrated 25 macro-processes (olive colour):

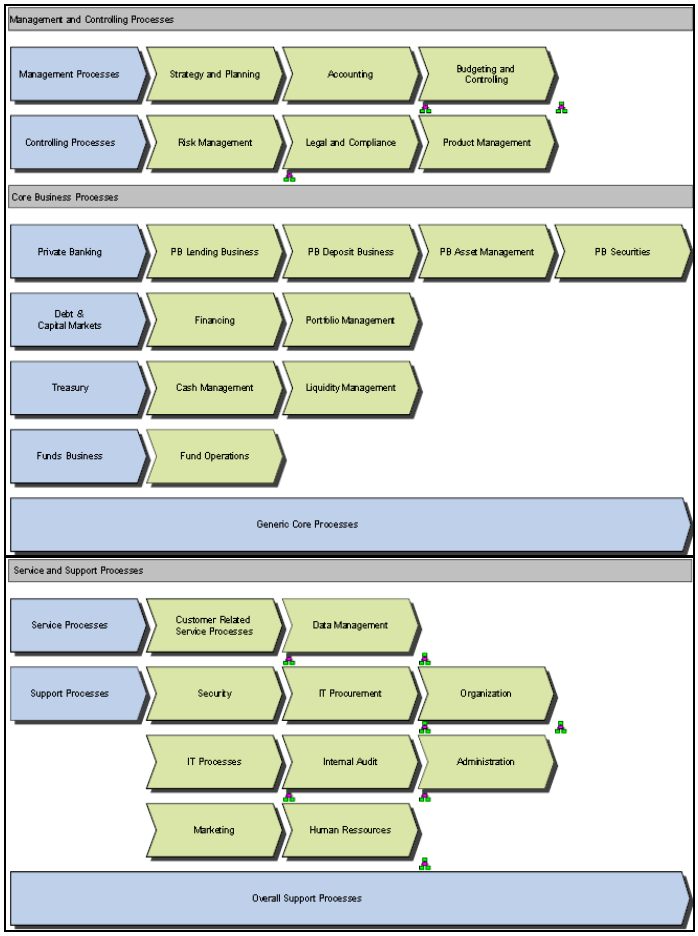



**Figure 74: Value Added Chain Model of Macro Processes LBBW**

With this overview, the scoping of processes is done. There are no relationships modelled between macro processes. "Data Management" is decomposed into further sub-processes which are Account Closure, Account Opening, Account Change etc. We exemplarily drill down into the "Account Closure" process, which is again a value added chain diagram:

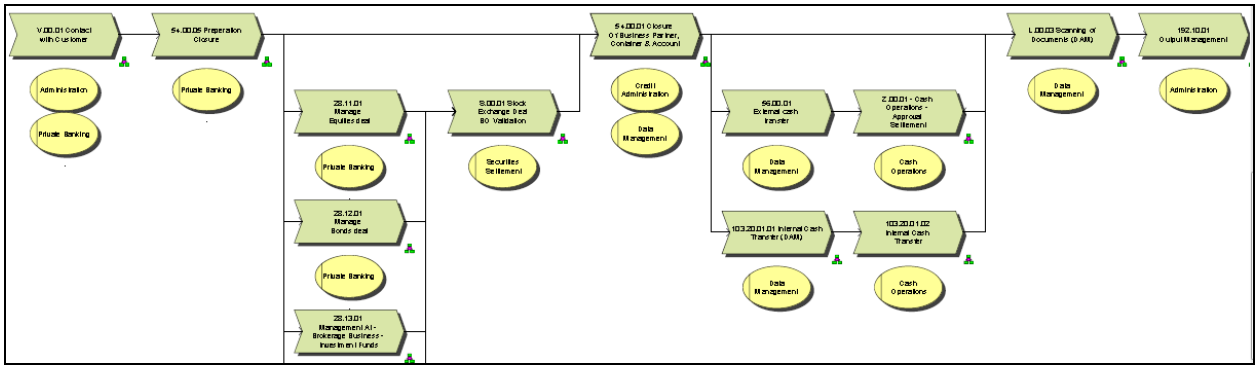


**Figure 75: Value Added Chain Model "Account Closure" Process LBBW**

The end-to-end process for “Account Closure “illustrates the processes necessary with sequential relationships and indication of organizational responsibility. This is done with yellow circle objects below the green process-objects.

#### 6.4.2.3. Fragment: AVE5 Service Oriented Business Process (Activity: Model Business Requirements with EPC)

The next fragment is about the modelling of business requirements with EPC. Exemplarily, we present the produced EPC model “Manage Equities Deal”. The same EPC conventions apply as introduced in figure 20 and 45 :

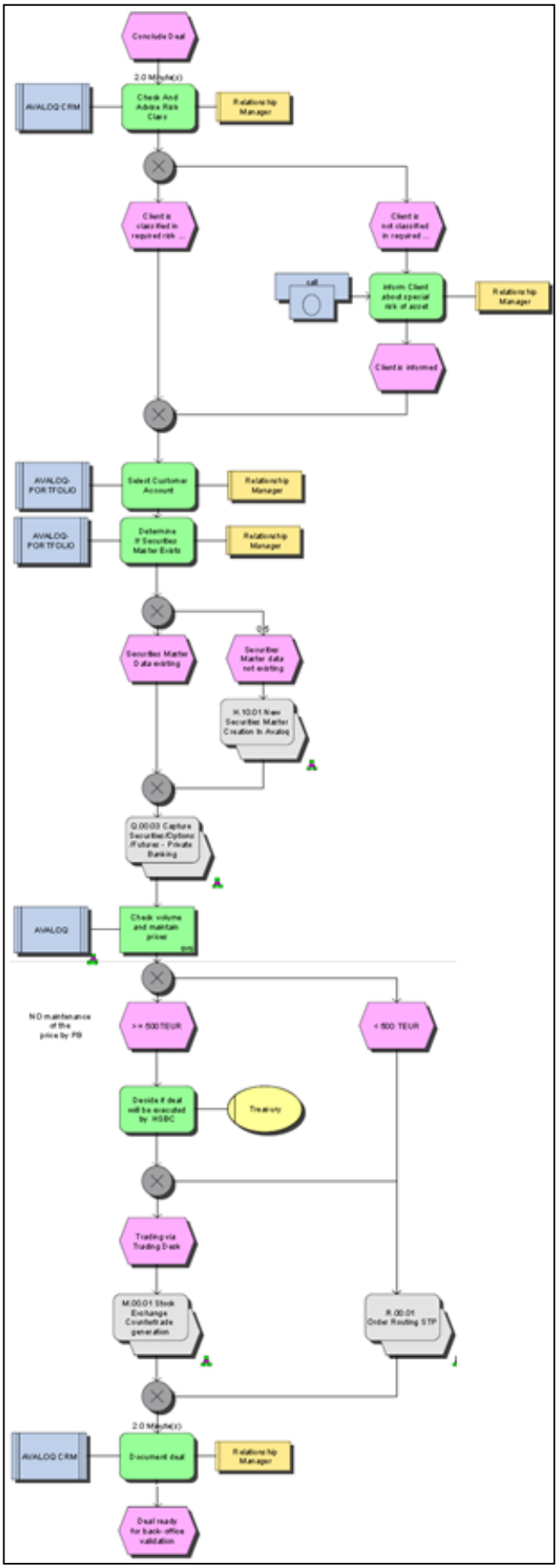



**Figure 76: EPC Process Modell Manage Equities Deal LBBW**

This process detail is showing the business requirements with business triggers (events), activities to perform and the related system service that is called to execute the activity. Overall, 180 use cases have been identified which could be decomposed into 500 processes. For each process, a target process has been created by following the EPC method fragment details.

#### 6.4.2.4. Fragment: LBBW1 Technical intermediary model (Activity: Model relationship between process, AVALOQ module and web-service)

For the activity e.g. "Check Volume and Maintain prices", a specific AVALOQ service is requested. The intermediary model (Access Diagram) is showing what service (web-service) is needed:

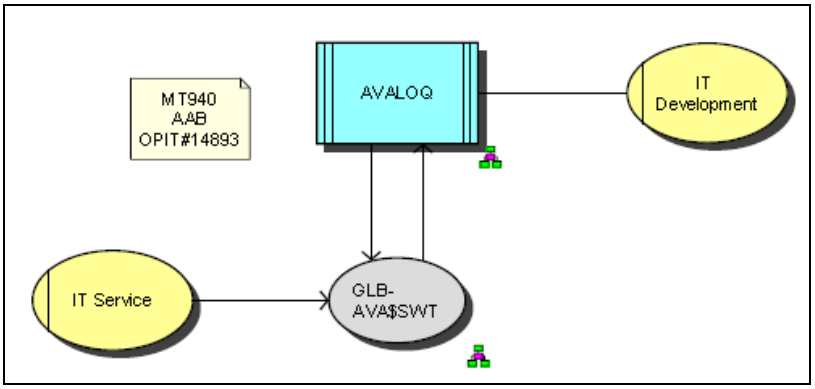



**Figure 77: Access Diagram AVALOQ/Web-Service LBBW**

IT Development organization is responsible for AVALOQ system, whereas IT Service department is responsible for the web-service "GLB-AVA$SWT". This fragment has been created only for LBBW purpose in order to formalize the relationship between process, AVALOQ module and web-service.

#### 6.4.2.5. Fragment: LBBW2 Web-Service Requirements (Activity: Model Web-Service Requirements)

The LBBW2 fragment again has been created for a specific purpose to model web-service requirements. A so called "AVALOQ Reference Database" proposed by the AVALOQ project team could not be used as descriptions of functionality were not process-oriented. Therefore, LBBW required detailed description of processes allowing technical implementation of the system. Exemplarily, the web-service object "GLB-AVA$SWT" can be further drilled down to get the technical web-service process description model. This model explains in EPC-notation, what exactly the web-service is supposed to do. It is indicating the platform, the technology and the conditions to perform the requested service (PIM). From there, the web-service is programmed in WSDL in order to implement and deploy it in the AVALOQ-system (PSM). To achieve this, the AVE5 fragment has been copied and modified in the method fragment database. The same conventions for EPC apply, but additionally also IS Services Objects and Information Carrier Objects are used:

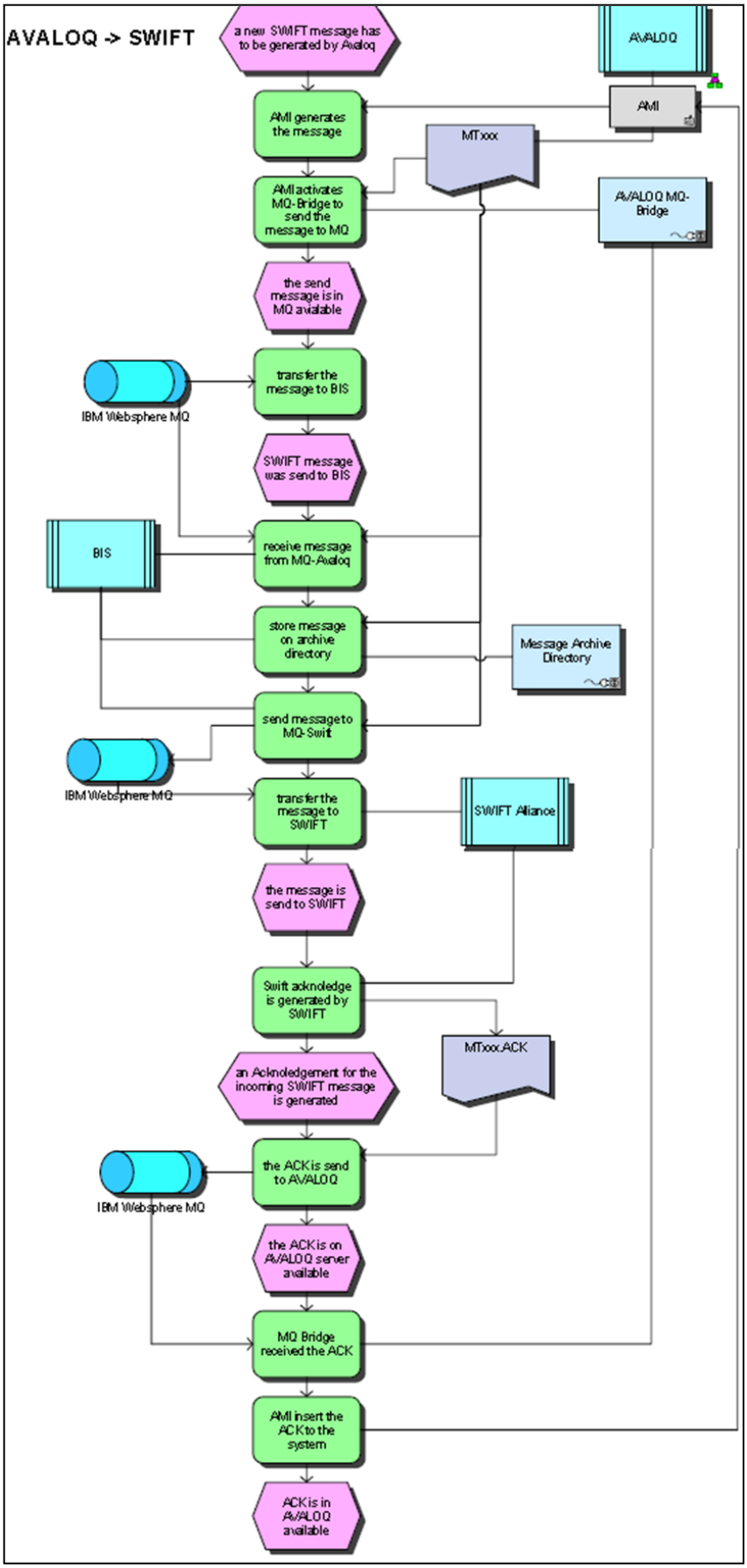



**Figure 78: Web-Service Requirement Model GLB-AVA$SWT LBBW**

## *6.5.* Conclusions of Field Trial Cases

The conclusion is structured into 3 parts: first, conclusions a made on validation discussion for the applied configuration process. Second, the outcome of the generated method fragments with the integration of modelling notations into the SOA Method and third, some conclusions on the applied field trial research method.

### 6.5.1. Conclusions on Validation Discussion of Configuration Process application

The main conclusions out of these cases can be summarized to following findings:

- the standard/best practice method proposed by industrial software provider and formalized into method fragments has not been used in its end-to-end application because the SOA methods were not adapted to individual choices of organizations and did not fit to situational context. In the first case, a mix between AVE SOA Method and SAP SOA Method was required (section 5.3.4.), whereas in the second case AVE SOA Method was used with additional self-created LBBW fragments (section 5.3.9.).
- the individual choices of organizations made for the selection of modelling languages were highly dependent on available IT application landscape, pre-defined IT application systems or knowledge of model designers.
- the IT run-time applications were selected because of business requirements matching and not because of SOA Method fit.
- therefore, the applied method must be situational specific adapted to IT landscape.
- the key question in every single implementation project is about the decision on where the cut is done between SOA design-time and SOA run-time and if there is a possibility to integrate both in the most effective and efficient way. This is highly dependent on interfacing capabilities of used tools and or applications. Guidance on these questions is represented in context and guidelines in the SOA Method Framework.

New was the utilization of EPC model to illustrate the details of web-services. On this deep level of detail, we would normally assume more technical models such as BPEL, UML sequence, state chart etc. but the chosen approach was successful. Hence, the method fragment of modeling EPC which is normally used on CIM level has been used on PIM level (LBBW2 Method Fragment), where it is normally not expected. AVE5 fragment has been copied and modified by LBBW and re-used for the PIM – Level, which was normally not foreseen in the description of the method fragment. This specific style of applying the method was a bit unconventional, but finally successful.

Overall the field trials showed successfully that existing SOA implementation methods are not good enough to be applied in the two real life field trial projects. In order to find a way to tailor the available methods to the situation-specific project, the decomposition of available methods into single fragments using an engineering method (section 4.1.) was successfully applied (section 6.3.). The validation discussion in chapter 6.3. Summarized feed-back on method application (section 6.3.5.1. and 6.3.10.1.) and satisfaction level with outcoming results (6.3.5.2. and 6.3.10.2.). The achieved method satisfaction by CIOs including also the project team showed clearly the benefit of the proposed approach.

Also without adhering strictly to the model-driven approach working with OMGs MDA rules, satisfactorily results were achieved. Contrarily to the state-of-the-art research on model-driven development guidelines such as MDA, model transformation has not been used in both cases. As said, the reasons for this were different and unfortunately, the impact on efficiency or time consumption using strict MDA instead could not be measured.

### 6.5.2. Conclusions on generated Method Fragment outcome

Related to the applied model fragment for selecting different modelling notations on different levels of detail, the following summary is showing the work products (natural language description for strategy not considered as modelling notation):

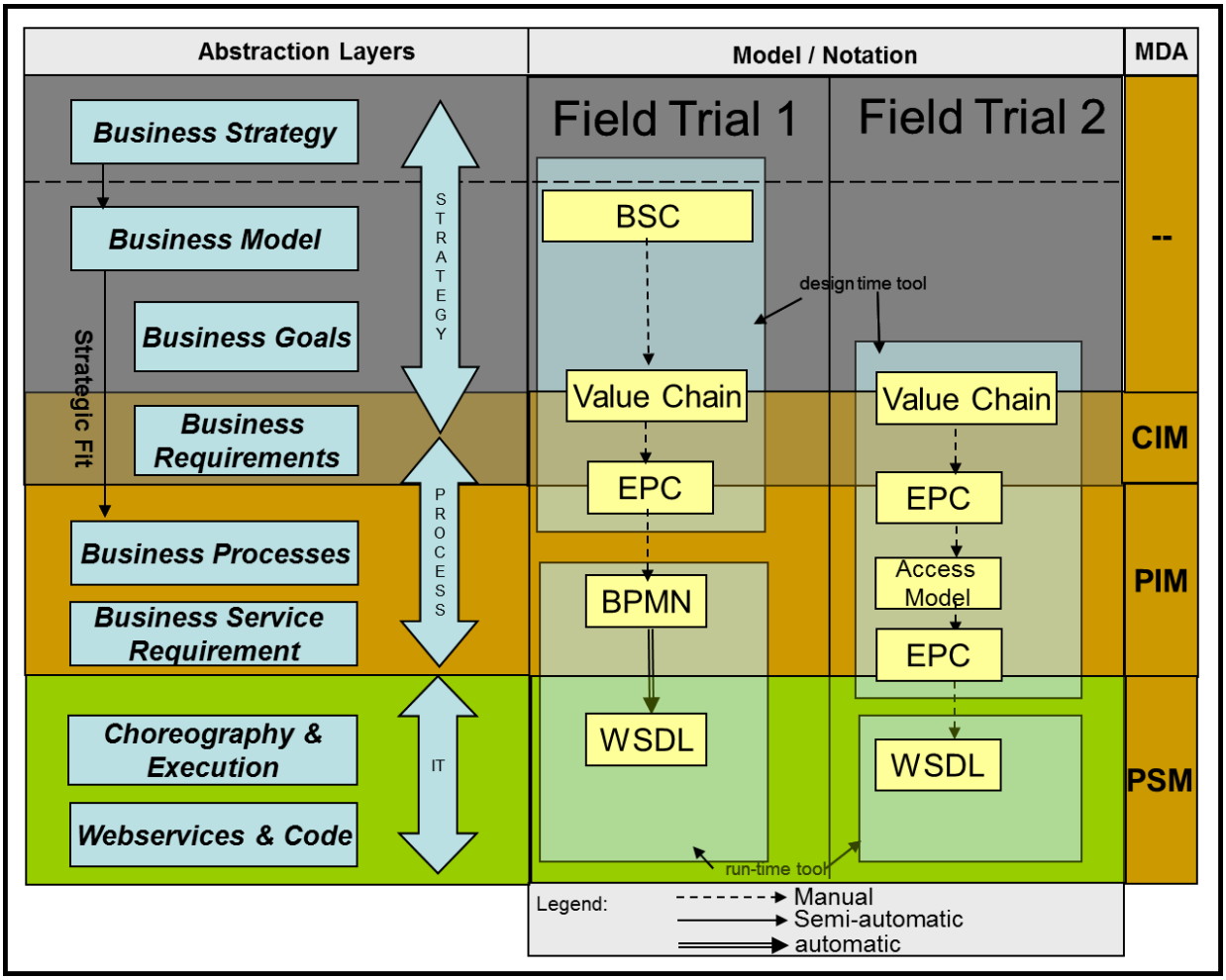



**Figure 79: Field trial Decision Summary of Modelling Language Path**

Field trial Cargolux formalized strategic Objectives in a BSC Model. For the high-level business process overview, Value Chain Model has been used. Then the link to EPC has been done to formalize business requirements. Then, the cut between SOA design time and SOA run-time was done. Technical BPMN models were created in SAP and enriched with WSDL web-service description.

The second case started with Value Chain Models for high-level understanding and scoping. Next, EPC has been used for business requirements. An Access Model (LBBW2) has been used to bridge EPC CIM (AVE4) to EPC PIM (LBBW2). The EPC on this level of detail was not expected, but has been successfully used to illustrate web-services description. These detailed web-service descriptions have been used to create WSDL descriptions in the run-time system AVALOQ.

### 6.5.3. Conclusions on applied field trial research method

Conclusion on the applied field trial method can be summarized to following remarks:

Generally, the validation by the field trials has shown that the Configuration Process for situational Method can be used in practice. Following to the field trial observations, the applied method could also work in other projects or practical cases.

The SOA projects were successful, but the positive impact of the applied method could not been measured with the field trial method.

Next, the field trial cannot be redone using another SOA Method or selecting other fragments. Each case with its context and constraints is unique which is creating uncertainty. Contrary to a laboratory experiment, the field trials cannot be redone nor trial context can be changed.

The influence of the author applying the configuration method in these trials related to the taken decisions for/against method fragments could not been measured.

# CHAPTER 7

# RESEARCH CONCLUSIONS, LIMITATIONS AND FUTURE WORK

---



---

This chapter is summarizing the research contributions (7.1.) related to the four research artefacts of the SOA Engineering Framework. First, the SOA Domain Model sets the conceptual frame (section 7.1.1.) for the SOA Method Qualification (section 7.1.2.). The configuration processes for situational SOA Method (section 7.1.3.) and the formalisation of SOA Method Fragments (7.1.4.) are four composing blocks of the SOA Method Engineering Framework. A conclusion on chosen viewpoints of top-down, model-driven and process-orientation (section 7.1.5.) is made. Next, identified limitations of current work (7.2.) are explained. Future work and areas of further development are highlighted (7.3.) and finally closing remarks is pointing to publications in relation with this thesis (7.4.).

## *7.1. Research Contributions*

During the last years, SOA reached more maturity and left the "hype"-topic area. New topics such as Software as a Service (SaaS), Process as a Service (PaaS), "Apps" and "Cloud" topics are attracting now more interest as "new" and upcoming technologies and process innovation.

Nevertheless, SOA is still on the agendas of organizations to explore the promised benefits the SOA paradigm can deliver. But to do so, it is required to apply a method which is adapted to the context and the situation in which the organization wants to apply the project. "One size fits all" for this particularly complex project type is not realistic. Consequently, the original research topic of a missing SOA engineering method has been addressed and a proposal taking the process-oriented and model-driven perspective has been shown through this thesis. This thesis has shown that available methods through state-of-the-art are not complete enough to cope with a consistent SOA engineering method. The existing SOA methods use specific views or are designed to accompany industrial software products. The method presented in this thesis is built on four main research contributions:

The first and second ones are the definition of the ***SOA Domain Model*** and the **qualification of existing SOA implementation method proposals** with focus on recommended SOA modeling languages on different levels of abstraction. Under the chosen viewpoint, the research question 1 (*Differences of available SOA Methods?)* has been investigated. Also

candidate modeling languages (research question 4: *which candidate modeling languages are suited to serve in SOA implementation?*) have been identified through the state-of-the-art.

The third main research contribution is the creation of **Configuration Process for SOA Situational Method** using **method engineering principles** linking to the SOA Domain Model particularly for the SOA Domain Modeling. This third research contribution investigates the research question about requirements for situated SOA methods (research question 2: *what is required for decomposing/recomposing a SOA Method?*). This question is also addressed through the fourth main research artifact about the formalization of **SOA Method Fragments** from available SOA Methods.

The application of these 4 contributions in real industrial field trial cases investigates the questions on practical decisions taken (research question 3: *what are decisions taken by organizations, when applying configuration process for SOA situational Method?)* on applying the method.

Next, as we are particularly interested in the SOA Modelling Domain, the research question 4 (research question 4: *Which candidate modeling languages are suited?*) and 5 (research question 5: *How to integrate the different kinds of modeling in a single SOA Method?*) are key. Finally, the overall satisfaction with the generated and applied situational configuration process for SOA and the satisfaction of result are investigated (research question 6: *what level of satisfaction for generated method achieved?* and *what is the level of satisfaction with situational SOA Method content outcome?*).

### 7.1.1. The SOA Domain Model

Based on literature review through Enterprise Architecture (section 2.1.), Modelling Languages (section 2.2.), Interfaces between abstraction layers (2.3.), Model transformation (section 2.4.) and the analysis of available SOA Methods/Frameworks (section 2.5.), the SOA Domain Model has been created. This SOA Domain Model is summarizing 5 domains (SOA Domain Modeling, SOA Domain BPM, SOA Domain Project, SOA Domain BPM Design Time & BPM Run-Time, SOA Domain Web Service). All domains and related sub-domains have been explained in detail to ensure common understanding and used semantics (section 3.1.). It can be considered as a condensed artefact, which can be used as a toolbox for different purposes. The SOA Domain Model has been introduced, explained and tested on completeness by industrial experts (section 3.3.). Furthermore, an excel-based guidance tool has been created to facilitate the SOA Domain Model application summarizing definitions and context descriptions for SOA sub-domains (section 5.4.). Based on questions or concerns to address, the user of the SOA Domain Model is enhanced. Consequently, the SOA Domain Model is a tool for identifying differences of available SOA Methods.

As we have chosen the viewpoint of modelling, a particular focus was made on best suited modelling languages for SOA implementation. Therefore, a complete overview model positioning the different modelling languages on related level of abstractions has been constructed (section 2.3 and 2.4.). The analysis of state-of-the-art followed by the validation of questionnaire participants (section 3.3.3.) has confirmed that specific modelling languages are more suited than others. The results have also confirmed that some modelling languages compete on the same level of detail. Additionally, transformation mechanisms between the levels of abstractions have been investigated.

### 7.1.2. SOA Method Qualification

The SOA Domain Model has been used to analyse and rate available SOA method proposals (section 3.2.). The qualification of SOA Methods has confirmed that SOA engineering principles are not applied and that currently available proposals are not adapted to the particular situation of organizations. The qualification tells nothing about the quality of the SOA Method, but more on the coverage of SOA Domains and SOA Sub-Domains. Depending on provenance of the SOA Method, the results vary a lot related to sub-domain coverage. The fact that SOA Methods are not adaptable to situation has also been expressed through the questionnaire results (section 3.3.3.) where the need for a situational SOA Engineering Method has been underlined.

### 7.1.3. Configuration Process for SOA Situational Method

Based on ME principles introduced in the state-of-the-art (section 2.6.), a SOA Engineering Method has been created, which is decomposed into 5 processes. These processes have been detailed (section 4.1.3.) and also the relationship between the SOA Domain Model and the Configuration Process through an alignment model has been explained (section 4.1.1.) With this, the requirements for situational SOA method were shown illustrating the decomposition and recomposition of SOA method with practical examples (section 4.2.1.).

The configuration process for SOA situational method has been applied and instantiated in the field trial cases (section 6.3.) to gather information on what concrete decisions practical organizations take. In both cases, valuable validation discussions (section 6.3.5.1 and 5.3.10.1.) on the application of the configuration process including the decisions taken for specific situations were made.

Furthermore, the quality of generated configuration process for SOA situational method has been evaluated (section 6.3.5.2. and 6.3.10.2.) and summarized (6.5.). It showed in both cases that the quality of generated method was satisfactorily because the applied method permitted consideration of individual situations.

Finally, the applied process with related decisions also indicated clearly suited modeling languages and how these languages were integrated in the SOA Method (figures 62, 67 and 79).

### 7.1.4. SOA Method Fragments

The SOA Method fragments were created (section 4.2.) from available SOA Methods (section 3.2.2. and 3.2.3.) To do this, ME principles have been used (section 2.6.) and a tooling support (section 5.3.) provided an efficient way of creating and storing method fragments in a fragment tool (Eclipse Process Framework) implementing SPEM 2.0. as defined by OMG.

These fragments were formalized and stored in the fragment database and were made available for re-use during the two concrete field trial project applications. Based on situational context, different fragments were selected (section 6.3.4. and 6.3.9.). In the LBBW case, 2 new fragments (LBBW1 and LBBW2) had to be created as available method

fragments could not satisfy the needs related to the specific situation. Generated method fragment details of these two field trials (section 6.4.) were explained and illustrated in detail including produced models. The conclusions on generated method fragment outcome (section 6.5.2.) were valuable as the selected and used modeling notations could be positioned on the different levels of abstractions and therefore closed the loop to the state-of-the-art and posed questions on candidate modeling languages and the integration into a single SOA Method.

The re-use of available method fragments showed exactly how organizations took individual method design decisions for the SOA implementation project. We have shown that both customized approaches were different from the standard SOA implementation methods. Consequently, it was necessary to assemble method fragments because of situational context.

These decisions were mainly driven by the SOA run-time operating system and the possibilities to interface with the design-time environment. The constructed method fragments are available in the method fragment database and can be re-used if situational context is similar or fitting.

Hence, through positive feedbacks of Cargolux and LBBW, the application cases showed therefore the value of the SOA Domain Model in relationship with the configuration process for situational SOA and the application of SOA Method Fragments.

### 7.1.5. Viewpoints of model-driven and process-oriented approach

The chosen viewpoints, as introduced in chapter 1 and detailed through the state-of-the-art in chapter 2, were also scoping this work and drawing the borderlines. The chosen viewpoints are of course not excluding or qualifying other viewpoints which could have been selected instead.

Following to the conducted world-wide questionnaire (section 3.3), nearly 77% of questionnaire respondents are using or have planned to use processes for the web-service identification and construction. Furthermore, the planned process-driven web service construction rate of 35,2 % is the highest value for the planned usage scenarios in BPM. Next, a process-oriented approach and process knowledge is rated as very important for SOA implementation by 90,7 % of respondents.

Both field trials used the same principles being top-down, model-driven and process-oriented. Modeling was used and process orientation was followed, but the MDA principles have not been applied to a full extent for good reasons. The applied approach with chosen modeling notations on the various levels of abstraction has shown in a consistent way, how these principles were used and what decisions were taken to implement a tailored and situational SOA implementation method.

## 7.2. Limitations

The SOA Domain Model is including a lot of sub-domains and related broad content to address and includes a wide range of topics. It is in this thesis not possible to detail all domains as deep as the SOA Modeling domain. This choice is necessary, as the thesis is taking the specific viewpoints as explained earlier. However, the other domains could be investigated in more detail through future work.

The questionnaire could not "validate" findings as statistically seen not enough participants responded to the questionnaire. As indicated already in the questionnaire limitations (section 3.3.), the statistical relevance with 54 valid respondents was unfortunately a bit low to conclude with empiric "validation" of asked questions.

The modeling language domain is very broad and de-facto completeness very difficult to achieve. Van der Aalst, ter Hofstede and Weske [vdAtHW03] were stating that an exhaustive list of modeling languages and comparing them was not possible. Consequently, the list was called "de-facto exhaustive" by summarizing state-of-the-art references to conclude with a long list of descriptive modeling languages. Another limitation to this consists in the difficulty to identify clearly and separate sharply quickly evolving modeling languages, notations, frameworks, meta-models and the degree of formalism.

SOA Implementation methods could only be evaluated on available information. The level of detail was varying between the methods and also related to the requirements and decisions within the methods. Proprietary methods from industrial vendors were sometimes not available to full extent and full detail because these were considered as a competitive advantage towards market competitors.

Method fragments were created taking specific choices (SPEM2.0. and Eclipse Process Framework Tool). It is not clear if other decisions would have led to other results.

The field trial cases were conducted only for 2 projects and two different industries and scope. The conclusions are only valid for these specific cases. The results can therefore hardly be generalized to many other cases. Finally, we conducted only a first cycle of the validation in the action cases. There could be more iterations helping to achieve another level of validation. The contribution to project success of using the proposed method could not be measured. It was not possible to extract the effect of using an efficient method against the effect of other criteria's could have e.g. "top management buy-in" or "skilled project team". Furthermore, it was not possible to redo the field trials and apply another method e.g. complete available "SOA Methods" or "pure technical approaches" being not process-oriented and model-driven to see if in that case the project would not have been successful or less successful.

## 7.3. Future Work

The SOA Domain Model tool could be enhanced by extending the model with content towards related SOA Sub-Domains from other or remaining SOA methods. Following to this, the SOA Domain Model could be updated (SOA Domain Modeling), enlarged (other SOA Domains) and similar application cases done in other SOA Model Domains.

The SOA Domain model is the baseline work for developing a complete ontology for SOA implementation. The first model with its method fragments examples could be refined and enlarged. Next, an increasing number of available method fragments e.g. with formalization of academic and industrial approaches in one database open for everybody could help to accelerate notoriety and finally also usage of the proposed situational engineering method.
Therefore, owners/creators of SOA methods would need to translate their methods into method fragments and post them into an online method fragment database. In reality, this is resource intensive and probably mostly interesting for consulting companies as they are selling SOA Implementation projects. As these consulting companies compete with each other, an open and shared method fragment database between various providers seems to be difficult.

Another area of work is to apply goal-driven analysis for the fragments in relationships to the criteria's. So far, there is a link between method fragments and SOA sub-domains, but there is no automatic mechanism e.g. such as dependency graphs who could evaluate automatically how well certain objectives are linked to these sub-domains. This research is ongoing [BCDV+11]. Next, other SOA Domains could be further formalized into fragments using the SOA alignment model (section 4.1.1.) between SOA Domain Model and ME terminology. This model so far cannot be generalized, as the mechanisms have been only applied on the SOA Domain "Modeling".

The guidance dimension is certainly also an area for future work. In particular, the guidance procedure on how to use the SOA Domain Model could be more enlarged to other implementation strategies such as meet-in-the-middle or bottom-up modeling. The best solution for this requirement might be a smart wizard tool proposing automatically different options based on requirements and project situation. Ideally, the SOA Domain Tool could be implemented into the Method Fragment Tool such as EPF. The configuration processes for situational SOA could be implemented into a workflow tool, which would ensure proper process execution enforcement.

The validation of proposed method could be further improved, by applying the method to case studies in bigger sized organizations outside Luxembourg with the objective of obtaining more feedback on practicability, strengths and weaknesses of the proposed SOA engineering Framework. Here, more iteration cycles could be done with the objective to achieve more robust evaluation and finally also possibilities to fine-tune the proposed framework.

The impact of the applied method on project success could be further investigated. So far, there is some work on success impact of deploying method (top-down, meet-in-the-middle, bottom-up) but not on exposing the impact of the situational SOA Implementation method to project success. This would be probably a thesis for its own, as a higher number of applications would be needed and a method to extract the effect in parallel projects to compare success rates.

## 7.4. Closing Remarks

The following publications related to this thesis are covering most of the scientific contribution:

[RP09] Ricken J., Petit M.: Requirements for BPM-SOA Methods: Results from an Empirical Study of Industrial Practice. Business Process Management Workshops 2009: 453-464, BPM2009, Springer, Volume 17, Part 8, 621-632, DOI: 10.1007/978-3-642-00328-8_62, Ulm, Germany, 2009

[Ric09] Ricken, J.: Results on Testing a SOA Domain Model through an empirical study – Executive Summary, Technical Report, University of Namur, Computer Science Faculty, Namur, Belgium, 2009.

[RP08] Ricken, J.; Petit M.: Characterization of Methods for Process-Oriented Engineering of SOA, in: Procedures of Collaborative Business Processes Workshop. BPM2008, Springer, DOI: 10.1007/978-3-642-00328-8_62 01.09 – 04.09.08, Milano, Italy, 2008

[Ric07]: Ricken J.: Top-Down Modeling Method for Model-Driven SOA Construction. OTM Workshops (1) 2007: 323-332 in: On the Move to meaningful Internet Systems (OTM): Volume 4805/2010, DOI: 10.1007/978-3-540-76888-3_54, Springer, LNCS, 2007

# BIBLIOGRAPHY

[ACDGK+03] Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., and Weerawarana, S.: Business Process Execution Language for Web Services, Version 1.1. Specification, BEA Systems, IBM Corp., Microsoft Corp., SAP AG, Siebel Systems, 2003
[AGAAG+08] Arsanjani A., Ghosh S., Allam A., Abdollah T., Ganapathy S., Holley K.: SOMA: A Method for developing service-oriented solutions, IBM Systems Journal 47 (3):377-396, 2008
[AHW03] van der Aalst, W.M.P., Hofstede, A.H.M., Weske, M.: Business Process Management: A Survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1-12. Springer, 2003
[Alo04] Alonso G., et al.: Web Service, Concepts, Architectures & Applications, Springer, Heidelberg 2004
[Ark02] Arkin, A.: Business Process Modeling Language (BPML). Spec., BPMI.org., 2002
[Ars04] Arsanjani A.: Service-Oriented Modelling and Architecture (SOMA), IBM developer-Works 2004
[ATHEN03] ATHENA: Deliverable DA 1.1.1.: First Version of State of the Art in Enterprise Modelling Techniques and Technologies to Support Enterprise InteroperabilityWork package – A1.1, 2003
[ATHEN05] Advanced Technologies for interoperability of Heterogeneous Enterprise Networks and their Applications: Deliverable D.A5.1: Perspectives in Service-Oriented Architectures and their Application in Environments that Require Solutions to be Planned and Customisable, Nov 2005
[ATHEN06] Advanced Technologies for interoperability of Heterogeneous Enterprise Networks and their Applications: Deliverable A6.3 : Model-driven and Adaptable Interoperability Framework, July 2006
[Bac00] Bachmann F. et al., Technical Concepts of Component Based Software Engineering, Technical Report, Carnegie-Mellon UniversityCMU/SEI-2000-TR-008 ESC-TR.-2000-007,2nd Edition, May 2000
[Bae92] Baets. W. Aligning information systems with business strategy. Journal of Strategic Information Systems,1(4), 1992.
[Bar03] Barry D.K : Web Services and Service Oriented Architectures, Morgan Kaufmann Publishers, 2003
[Bar05] Barnes M.: Benefits and Challenges of SOA in Business Terms, Gartner Whitepaper G00130078, Sept. 2005
[BCDV+11] Birkhölzer T., Chiniforoshan H., Dickmann C., Vaupel J., Ast S.: Goal-Driven Evaluation of Process Fragments Using Weighted Dependency Graphs, ICSSP 11, Honolulu, USA, 2011.
[BD07] Berson A., Dubov L.: Master Data Management and Customer Data Integration for a Global Enterprise, MCGraw-Hill, 2007
[Ben87] Benbasat, I. et al.: The Case Research Strategy in Studies of Information Systems, MIS Quartely, Vol 11, No 3, pp369-386, 1987
[Ber08] Berkem B.: "From BMM to SOA", in Journal of Object Technology, vol. 7, no. 8, November –December 2008, pp. 57-70 http://www.jot.fm/issues/issue_2008_11/column6/
[BGBDK+05] Bourey J.-P., Grangel R., Berre A., Doumeingts G., Kalampoukas K., Bertoni M., PondrelliL., Daclin N.: Report on Model Establishment, Deliverable DTG2.1, Interoperability Research for Networked Enterprises Applications and Software, Network of Excellence (INTEROP), Contract no.: IST-508 011 (2005)

[BGLOS+09] Blinco K., Grisby T., Laird A., O'Neill O., Srikanth V., and Smythe C.: Adoption of service oriented architecture (SOA) for enterprise systems in education: Recommended practices.Technical report, IMS Global Learning Consortium, 2009
[BGPPW+10] Buckow H., Gross HJ., Piller G., Prott K., Willkomm J., Zimmermann A.: Analyzing the SOA Ability of Standard Software Packages with a dedicated Architecture Maturity Framework: in Klink, Koschmider, Mevius, Oberweis (Hrsg): Proceedings: EMISA 2010: Einflussfaktoren auf die Entwicklung flexibler, integrierter Informationssysteme, Karlsruhe 07.10-08.10.2010
[BGR07] Bandara W., Gable G., Rosemann M. : Critical Success Factors of Business Process Modeling, QUT e_prints, 2007, http://eprints.qut.edu.au/8755/1/8755.pdf
[BHABT+04] Berre, A-J., Hahn, A., Akehurst, D., Bezivin, J., Tsalgatidou, A., Vermaut, F., Kutvonen, L. and Linington, P. : Deliverable D9.1: 'State-of-the-art for interoperability architecture approaches', EU INTEROP Network of Excellence., 2004
[BL06] Benguria, G., Larrucea, X. et al.: A Platform Independent Model for Service Oriented Architectures, I-ESA conference 2006, March 22-24, Bordeaux, Springer
[Blai05] Blaikie, N.: Designing Social Research, Cambridge, UK, Polity Press, 2005
[BLW96] Brinkkemper S., Lyytinen K, Welke R.: Method engineering: principles of method construction and tool support : proceedings of the IFIP TC8, WG8.1/8.2 Working Conference on Method Engineering 26-28 August 1996, Atlanta, USA. Springer. ISBN 041279750X
[BMR+96] Buschmann F., Meunier R., Rohnert H., Sommerlad P., Stal M.: Pattern-oriented Software Architecture - A System of Patterns. J. Wiley and Sons Ltd., 1996.
[BN94] Bernus P., Nemec L.: CSIRO Division of Manufacturing Technology, Manufacturing Technology Management Research Report # MTM366, 1994, http://www.cit.gu.edu.au/~bernus/taskforce/geram/report.v1/report/report.html
[BNSSW+04] Bradburn, Norman M., Sudman, Seymour, Wansink, Brian, Asking Questions: The Definitive Guide to Questionnaire Design – For Market Research, Political Polls, and Social and Health Questionnaires. Jossey-Bass. 2004
[BPDM08] Business Process Definition Metamodel, OMG Spec Version 1.0. published 3.11.2008, http://www.omg.org/spec/BPDM/1.0/volume1/PDF/
[BPTRE09] http://www.bptrends.com/, Website, 2007
[BR10a] Brocke, J. & Rosemann, M.: Handbook on Business Process Management 1: Strategic Alignment, Governance, People and Culture, International Handbooks on Information Systems, Springer, Vol. 1, 2010
[BR10b] Brocke, J. & Rosemann, M.: Handbook on Business Process Management 2: Strategic Alignment, Governance, People and Culture, International Handbooks on Information Systems, Springer, Vol. 2, 2010
[Bri06] Brittenham, P.: Web-Services Development Concepts, IBM Software Group, http://www-06.ibm.com/software/solutions/webservices/
[Bri96] Brinkkemper S, : Method Engineering : engineering of information systems development methods and tools, Information & Software Technology, Elsevier Science, 1996
[BRH99] Brinkkemper S., Saeki M., Harmsen F.: Meta-modelling based assembly techniques for situational method engineering, Information Systems, Volume 24, Issue 3, 10th International Conference on Advanced Information Systems Engineering, May 1999, Pages 209-228
[Bue07] Buecker A. et al: IBM: Understanding SOA Security: Design & Implementation, RedBooks, http://www.redbooks.ibm.com/redbooks/pdfs/sg247310.pdf, Nov 2007
[But09] Butler J. et al in CBDI Journal: CBDI-SEA SOA Reference Framework, http://www.cbdiforum.com/secure/interact/2007-03/the_architecture_component.php, retrieval 15.12.2009
[C4IS97] C4ISR Architecture Working Group (1997), US Department of Defense, 18.12.1997, http://www.c3i.osd.mil.org/cio/i3/AWG_digital_library/pdfdocs/fw.pdf

[CAR09] Cargolux, www.cargolux.com
[CAR11] Cargolux Annual Report 2011
[CBDI09]: CBDI-SAE Meta Model For SOA Version 3, http://everware-cbdi.com/mm-v3
[Cen90] CEN ENV 40 003: Computer–Integrated Manufacturing – Systems Architecture Framework for Enterprise Modelling, CEN/CENELEC, Brussels, Jan 1990
[Cen95] CEN ENV 12204: Computer–Integrated Manufacturing – Constructs for Enterprise Modelling, CEN/CENELEC, Brussels, December 1995
[CH06] Chanliau M., Hynes D.: SOA Governance: What's Required To Govern And Manage A Service-Oriented Architecture ORACLE Whitepaper , October 2006 http://www.oracle.com/technology/tech/standards/pdf/governance.pdf
[Cha03] Channabasavaiah K. et al, : Migrating to Service Oriented Architecture – part 1, IBM Developer Works, 2003
[Chi04] Chinnici R. et al.: Web Service Description Language (WSDL) 2.0., March 2004, http://www.w3.org/TR/wsdl20-soap11-binding/
[Che76] Chen P.: "The Entity-Relationship Model: Toward a Unified View of Data" , ACM on Database Systems, Vol. 1, No. 1, 1976
[Cre98] Creswell, J. : Qualitative inquiry and research design: Choosing among five traditions. Thousand Oaks, California: Sage Publications, 1998
[DDDG08] Decker G., Dijkman R., Dumas M, Garcia-Banuelos L.: Transforming BPMN Diagrams into YAWL Nets, in Proceedings of Dumas, Reichert Ming-Chien: Business Process Management: 6th International Conference, Milan, Sept. 2008
[DG06] Derzsi Z. and Gordijn J.: A framework for business/it alignment in networked value constellations. In T. Latour and M. Petit, editors, Proceedings of the workshops of the 18th CAiSE conference, pages 219–226, Namur, B, 2006. Namur University Press.
[Dos05] Dostal W. et al : Service-orientierte Architekturen mit Web Services: Konzepte – Standards – Praxis. Heidelberg; München : Elsevier, Spektrum Akad. Verl., 2005
[Dou07] Doumeingts G. et al, Report on Model Establishment, Deliverable DTG2.1, INTEROP (Interoperability Research for Networked Enterprises Applications and Software, Network of Excellence - Contract no.: IST-508 011), 2007
[DP07] Dubois E., Pigneur Y. et al.: Deliverable DTG5 : Final Report on the comparison/mapping/evolution between business models, INTEROP , April 2007
[DvdA04] Dehnert, J., van der Aalst, W.M.P.: Bridging Gap between Business Models and Workflow Specifications. International Journal of Cooperative Information Systems, 2004
[EBFG+02] van Eck, P.,, Blank, H., Fokkinga, M., Grefen, P., Wieringa, R.: A Conceptual Framew for Architecture Alignment Guidelines. http://graal.ewi.utwente.nl/GRAAL_whitepaper_20021017.pdf (2002)
[Eclipse09] http://www.eclipse.org/epf/
[EG02] Eder J., Gruber W.: A Meta Model for Structured Workflows Supporting Workflow Transformations. ADBIS 2002: 326-339
[EJ08] Edirisuriya A., Johannesson P.; On the Alignment of Business Models and Process Models, Workshop Processing BPM2008 in Business Process Design, Milano, Italy Springer 2008
[Erl05] Erl T.: Service Oriented Architecture: Concepts, Technology and Design, 2006
[FHLNH+98]: Falkenberg ED., Hesse W., Lindgren P., Nielsson B., Oei H., Rolland C., Stamper RK., van Asche F., Verrijn-Stuart S., Voss A.: A Framework of Information System Concepts (FRISCO), Computer Science Department University of Leiden, Netherlands, 1998
[Fow02] Fowler M.: Patterns of Enterprise Application Architecture. Addison-Wesley, 2002.
[Fox92] Fox, M.S.: The TOVE Project, Towards a Common Sense Model of the Enterprise," in Enterprise Integration, Ch. Petrie (Ed), Cambridge, MA, MIT Press (1992)
[Fra03] Frankel D.: The Zachman Framework and the OMG's Model Driven Architecture, Whitepaper, Sept 2003

[GAJV08] Gottschalk F, van derAalst W. M. P., Jansen-Vullers W. M. P., Verbeek H. M.W.:Protos2CPN: Using colored Petri nets for configuring and testing business processes. In: International Journal for Software Tools for Technology Transfer (STTT) 10, Springer, 2008
[Gar03] Gardner, T.: UML Modelling of Automated Business Processes with a Mapping to BPEL4WS. In Proceedings of the First European Workshop on Object Orientation and Web Services at ECOOP 2003.
[GART06] The Gartner SOA Adoption Model, GARTNER Leading Toolkit, December 2006
[GFFCM+04]Groves, Fowler, Floyd, Couper, Mick, Lepkowski, Singer, Eleanor, Tourangeau : Questionnaire Methodology, John Wiley & Sons, Inc. 2004
[GAR11] Gartner: Magic Quadrant for Business Analysis Tools, G00219247, Published: 12.12.2011
[GHJV94] Gamma E., Helm R., Johnson R., Vlissides J.: Design Patterns: Elements of Reusable Object-Oriented Software.
[GPHS08] Gonzales-Perez C., Henderson-Sellers B: Meta Modelling for Software Engineering, Wiley, 2008
[GPW06] Gordijn J, Petit M., Wieringa R: Understanding business strategies of networked value constellations using goal and value modeling, In: 14th IEEE International Requirements Engineering Conference (RE'06), 11-15 September, Minneapolis
[GPZ10] Giannoulis C., Petit M., Zadravkovic J.: Towards a Unified Business Strategy LAnguage : A Meta-Model of Strategy Maps, in: P.van Bommel et al (Eds): PoEM 2010, LNBIP 68, pp. 205-216, 2010
[Gru04] Gruber T.: Interview with Tom Gruber: Bulletin of AIS Special Interest Group on Semantic Web and Information Systems (SIGSEMIS), 1(3), 2004
[Gru93] Gruber T.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In N. Guarino and R. Poli, editors, Formal Ontology in Conceptual Analysis and Knowledge Representation, Deventer, The Netherlands, 1993.
[Gün05] Güner S.: Service Oriented Architecture, Master Thesis, University of Hamburg, 2005
[GP11] Greefhorst D., Proper E.: Architecture Principles, 1st Edition SPRINGER, 2011
[HA06] Horner, J.; Atwood, M.E.: Effective Design Rationale: Understanding the Barriers", in Dutoit, A.H.; McCall, R.; Mistrík, I. et al., Rationale Management in Software Engineering, Springer Berlin Heidelberg, 2006
[Har97] Harmsen, A.F.: Situational Method Engineering, PhD Thesis, University of Twente, The Netherlands, January 1997
[HBO94] Harmsen, A.F., Brinkkemper, S., Oei, H.: Situational Method Engineering for Information Systems Projects. In Methods and Associated Tools for the Information Systems Life Cycle. Proceedings of the IFIP WG8.1 Working Conference Cris/94, T.W. Olle, A.A. Verrijn-Stuart, Eds. North Holland, Amsterdam, 1994, 169-194.
[HH95] Hoef, R., Harmsen, F. : Quality Requirements for Situational methods. In: Grosz, G. (Ed.) In proceedings of the Sixth Workshop on the next generation of CASE tools, Jyväskylä, Finland, June 1995
[Hid93] Hidding G. et al: Modeling large processes with task packages, workshop on modeling in the Large, AAAI conference, Washington DC, 1993
[Hil00] Hilliard,R. , IEEE-Std-1471-2000 IEEE-Std-1471-2000 Recommended Practice for Architectural Description of Architectural Description of Software-Intensive Systems Software-Intensive Systems, http://www.enterprise architecture.info/Images/Documents/IEEE%201471-2000.pdf, 2000, date of download 1.7.07
[HO93] Heym M, Oesterle H: Computer-aided methodengineering, in Information and Software Technology Vol 35 No 6/7 pp345-354, 1993

[Hol04] Hollingsworth, D.: The Workflow Handbook 2004,chapter The Workflow Reference Model: 10 Years On, pages 295{312. Workflow Management Coalition, 2004
[HR00]: Hommes BJ., van Reijswoud V.: Assessing the Quality of Business Process Modelling Techniques in Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000
[HR08] Helms, R.W., Reijsen, J.: Impact of Knowledge Network Structure on Group Performance of Knowledge Workers in a Product Software Company. In D. Harorimana & D. Watkins (Eds.), Proceedings of the 9th European Conference on Knowledge Management (pp. 289-296). UK: Academic Conferences, 2008
[HS96] Harmsen F.Saeki M: "Comparison of four method engineering languages". In: Sjaak Brinkkemper et al. (eds.) Proceedings of the IFIP TC8, WG8.1/8.2 working conference on method engineering on Method engineering : principles of method construction and tool support: principles of method construction and tool support, 1996
[HSR10] Henderson-Sellers, B., Ralyté, J. : Situational Method Engineering: State-of-the-Art Review. Journal of Universal Computer Science, Vol. 16, no. 3 2010, pp. 424-478
[HV93] Henderson J and Venkatraman N., Strategic Alignment: Leveraging Information Technology for transforming organisations - IBM Systems Journal, vol32, NO1, 1993
[HW03] Hohpe G., Woolf B.: Enterprise Integration Patterns. Addison-Wesley, 2003.
[HZ06] Hentrich C., Zdun U.: Patterns for process-oriented integration in service-oriented architectures. In Proceedings of 11th European Conference on Pattern Languages of Programs (EuroPLoP 2006), Irsee, Germany, July 2006.
[IA07] Inaganti I., Aravamudan S.: The SOA Maturity Model, in BPTrends publication, April 2007,
http://www.bptrends.com/publicationfiles/04%2D07%2DART%2DThe%20SOA%20MaturityModel%2DInagantifinal%2Epdf.
[IBMSS03] IBM, BEA Systems, Microso., SAP AG, and Siebel Systems. Business Process Execution Language forWeb Services.
ftp://www6.software.ibm.com/software/developer/library/ws-bpel.
pdf, 2003
[IEE00] IEEE. Recommended Practice for Architectural Description of Software Intensive Systems. Technical Report IEEE-std-1471-2000, IEEE, 2000.
[IEEE12] IEEE Explore: http://ieeexplore.ieee.org/search/advsearch.jsp
[IG04] Iyer, B., Gottlieb, R.: The Four-Domain-Architecture: An approach to support enterprise architecture design. IBM Systems Journal 43) ((2004) 587–597
[ISO00a] International Organization for Standardization: RM ODP - Open Distributed Processing
[ISO00b] International Standards Organization (ISO), 15704:2000 Industrial automation systems -- Requirements for enterprise-reference architectures and methodologies, 2000,
[ISO01] International Standards Organization (ISO), ISO/IEC 9126-1:2001,Software Quality Attributes, Software Engineering – Product Quality, Part 1: Quality Model, 2001.
[ISO06a] International Standards Organization, ISO: ISO 19436-2006: Enterprise Integration, Framework for Enterprise Modeling, 2006
[ISO06b] International Standards Organization, ISO: ISO 19439:2006 Enterprise integration -- Framework for enterprise modelling, 2006
[ISO10]: Survey of Architecture Frameworks: http://www.iso-architecture.org/ieee-1471/afs/frameworks-table.html
[ISO11]: ISO/IEC/IEEE 42010: Systems and Software Engineering – Architecture Description, 2011
[ITNAT09] www.itnation.eu, Website, 2009
[JB96] Jablonski, S.; Bußler, C.: Workflow Management Modeling: Concepts, Architecture and Implementation, Thomson, 1996.

[JBR99] Jacobson I, Booch G, Rumbaugh J.: The Unified Software Development Process, Adiso-Wesley, Reading, Massachusetts, 1999
[JBS97] Jablonski , M. Boehm M, Schulze W.(Hrsg.): Workflow-Management: Entwicklung von Anwendungen und Systemen - Facetten einer neuen Technologie. dpunkt Verlag. April 1997
[JC99] Jorgensen H., Carlsen S.: Emergent workflow: Integrated planning and performance of process instances. In Proc. Of Workflow Management' 99. Muenster, Germany, 1999.
[Jic79] Jick, T.D.: Mixing qualitative and quantitative Methods: Triangulation in Action, Administrative Science Quarterly, Vol 24. No.4.pp.602-611, 1979
[JK04] Jost W., Kruppke H.: Business Process Management: der ARIS Value Engineering-Ansatz, in Innovation durch Geschäftsprozessmanagement: Scheer, Abdolhassan, Kruppke, Jost, p. 15-23, Springer, 2004.
[JLS92] [Jarczyk, A., Löffler, P., Shipman F.. : "Design Rationale for Software Engineering: A Survey", 25th Hawaii International Conference on System Sciences, 1992]
[Joh05] Johnston S.: Modeling Service Oriented Solutions, IBM developer works, July 2005, http://www.ibm.com/developerworks/rational/library/jul05/johnston/index.html.
[Jon02] Jonkers H.: State of the Art in Architecture Concepts and Description, Archimate Deliverable D 2.1., 20.12.2002
[Jon05] Jones S.: A methodfor service Architectures, CapGemini, October 2005, http://glintech.com/downloads/A%20Methodology%20for%20Service%20Architectures.pdf
[JV90] Joryz, R. H. and Vernadat, F. B.,: CIM-OSA Part I: Total Enterprise Modeling and Function View," International Journal of Computer Integrated Manufacturing, Vol.3, No. 3, pp. 144-156., 1990
[KBS06] Krafzig, D.; Banke, K.; Slama, D. : Enterprise SOA : service-oriented architecture best practices. 5. print. Upper Saddle River [u. a.] : Prentice Hall PTR, 2006
[Ken02] Kent S.: Model Driven Engineering, Integrated Formal Methods pages 286-298, Springer 2002
[KG07] Kort C., Gordijn J.: Modeling strategic partnerships using the e3-value ontology: A field study in the banking industry, 2007
[Kin04] Kindler E.: On the semantics of EPCs: A framework for resolving the vicious circle. In Business Process Management, pages 82-97, 2004.
[KJS96] Karagiannis D., Junginger S., Strobl R.: Introduction to Business Process Management System Concepts, in: B. Scholz-Reiter, E. Stickel (Eds.): Business Process Modelling, Lecture Notes in Computer Science, 1996, Springer.
[KL03] Kavakli E., Loucopoulos P.: Goal Driven Requirements Engineering: Evaluation of current methods, 8th CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD '03), 2003]
[KLR96] Kelly S. Lyytinen K., Rossi M.: MetaEdit+ A fully Configurable Multi-User and Multo-Tool CASE and CAME Environment, Lecture Notes in Computer Science: Advanced Information Systems Engineering, Springer 1996
[Klu07]: Klueckmann J.: 10 Steps to Business Driven SOA, IDS Scheer Expert Paper, 2007
[KN92] Kaplan R, Norton D.: The Balanced Scorecard – Measures That drive Performance, Harvard Business Review, jan/feb: pp71-79, 1992
[KN93] Kaplan R, Norton D.: Putting the Balanced Scorecard to work. In: Harvard Business Review. 1993
[KN04a] Kaplan R, Norton D.: The Strategy Map: Converting Intangible Assets into Tangible Outcomes, Harvard Business School Press, Boston, 2004
[KN04b] Kaplan R, Norton D.: The Strategy Map: Guide to aligning intangible assets. J Staretgy & Leadership 32, 10-17, 2004

[KNS92] Keller, G., Nuettgens, M., and Scheer, A. W.: Semantische Prozessmodellierung auf der Grundlage \Ereignisgesteuerter Prozessketten (EPK)". Heft 89, Inst. für Wirtschaftsinformatik, SaarbrÄucken, Germany Report BPM-05-13, BPMcenter.org., 1992
[KR70] Kunz, W., Rittel, H.: Issues as elements of information systems. Working Paper 131, Center for Urban and Regional Development, University of California Berkley, 1970
[Kru95] Kruchten P., The 4+1 View Model of Architecture, IEEE Software, vol 12, no.6, pp.42-50, 1995
[KW92] Kumar K, Welke RJ : Method Engineering: a proposal for situation specific methodconstruction, in Cotterman: Challenges and Strategies for Research Systems Development, John Wiley, 1992
[KWB03] Kleppe A., Warmer J., Bast W.: MDA explained: the model driven architecture: practice and promise. Addison Wesley Professional, Boston, 2003.
[Lan05] Lanckhorst M. et al : Enterprise Architecture at Work, Springer Verlag, 2005
[LBBW09a]Landesbank Rheinland Pfalz / Landesbank Baden-Würtemberg: http://www.lbbw.lu/index.htm
[LBBW09b]Landesbank Rheinland Pfalz / Landesbank Baden-Würtemberg: Presentation of the AVALOQ Project, Pocess World Munich, 2009
[LBBW12] Year end Closing Report/Einzelabschluss nach IFRS 2011, 2012 www.lbbw.lu
[LDB09] de Leusse P., Dimitrakos T., Brossard D.: "A Governance Model for SOA," icws, pp.1020-1027, 2009 IEEE International Conference on Web Services, 2009
[Lee05] Lee J. : Architecting Industry Standards for Service Orientation, Microsoft, 2005, http://msdn2.microsoft.com/en-us/library/ms978270(d=printer).aspx
[Lee97] Lee, J. (1997). "Design Rationale Systems: Understanding the Issues". IEEE Expert 12, 1997
[LK06] Lehtola, L., Kauppinen, M.: Suitability of requirements prioritization methods for market-driven software product development. Software Process: Improvement and Practice, 2006
[LMW05] Lankes J., Matthes F., Wittenburg A.: Architekturbeschreibung von Anwendungslandschaften: Softwarekartographie und IEEE Std 1471-2000, Whitepaper, http://wwwmatthes.informatik.tu-muenchen.de/file/Publikationen/2005/LMW05b/041214-LaMaWi-SE-Paper.pdf, 2005
[LPB95] Luftman J., Papp R., Brier T.: The strategic alignment model: Assessment and validation. In Proceedings of the Information Technology Management Group of the Association of Management (AoM) 13th annual International Conference, pages 57–66, Vancouver, 1995.
[LPWCS09] Land M., Proper E., Waage M., Cloo J., Steghuis C.: Enterprise Architecture: Creating Value by Informed Governance, Springer, 2009
[LRI12] LRI Invest Luxembourg, www.lri-invest.de
[Lub07] Lublinski B.: Service Oriented Decomposition to meet architectural goals, http://www.ibm.com/developerworks/library/ar-soadecomp/, Oct 2007
[Mac06] MacKenzie C.M. et al: Reference Model for Service Oriented Architecture 1.0, OASIS, February 2006
[MAL98] Mintzberg H., Ahlstrand B. and Lampel J.: A Guided Tour Through the Wilds of Strategic Management, FT Prentice Hall, 2002
[MB97] Mingers, J., Brocklesby J.: Multimethodology: Towards a framework for Mixing Methods, International Journal of Management Science, Vol 25, No.5, pp. 489-509, 1997
[MCFKP+95] Mayer R , Crump J., Fernandes R., Keen A., Painter M. : Information Integration for Concurrent Engineering (IICE) Compendium of methods report Air Force Material Command, Wright-Patterson Air Force Base, Ohio. 1995 p.7-10.,, http://www.idef.com/pdf/compendium.pdf,

[MCvG05] Mens T., Czarnecki K. , Van Gorp P.: A Taxonomy of Model Transformations, in “Language Engineering for Model-Driven Software Development". Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, 2005
[MLZ05] Mendling J., Lassen K. B., Zdun U.: Transformation strategies between blockoriented and graph-oriented process modelling languages. Technical Report JM- 200510 -10, WU Vienna, 2005.
[MLZ06] Mendling, J., Lassen, K.B., Zdun, U.: `On the Transformation of Control Flow between Block-Oriented and Graph-Oriented Process Modeling Languages', Int. J. Business Process Integration and Management, 2006
[MNN04] Mendling, J., Nuttgens, M., and Neumann, G: A Comparison of XML Interchange Formats for Business Process Modelling. In Feltz, F., Oberweis, A., and Otjacques, B., editors, Proceedings of EMISA 2004, LNI 56, pages 129{140. 25, 2002, Version 1.0, Workflow Management Coalition, 2004
[MS95] March S.T., Smith, G.F.: Design and natural science research on information technology, Decision Support Systems vol. 15, pp. 251-266, 1995.
[MZ05] Mendling J., Ziemann J.: Transformation of BPEL processes to EPCs. In Proc. of the 4th GI Workshop on Event-Driven Process Chains (EPK 2005), volume 167, pages 41–53, Dec 2005.
[NEKZ+05]: Nurcan S., Etien A., Kaabi R., Zoukar I., Rolland C.: A strategy driven business process modelling approach, Business Process Management 11, pp 628-649, 2005
[NK06]: Nysetvold AG., Krogstie J.: Assessing Business Processing Modelling Languages Using a Generic Quality Framework in Advancd Research in Database Topics Vol5 (Keng Siau), Idea Group Publishing, 2006
[NL05] Newcomer, E.; Lomow, G. : Understanding SOA with Web services. Upper Saddle River : Addison-Wesley, 2005
[NvdHP+09] Nguyen, D.K., Van den Heuvel W.J., Papazoglou M.P., Castro V., Marcos E.: Gap Analysis Methodology for Business Service engineering. In Proceedings of 11th IEEE Commerce and enterprise Computing (CEC’09), IEEE, Vienna, 2009
[OASIS06] OASIS, Reference Model for SOA 1.0., February 2006
[OASIS07] OASIS. Business Process Execution Language (WSBPEL) 2.0. http://docs.oasisopen. org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf, May 2007.
[Oll91] Olle T et al.: Information Systems Methods – a Framework for Understanding, Addison-Wesley, 1991
[OMG]: Unified Modeling Language: http://www.omg.org
[OMG02] OMG. Model-driven Architecture. http://www.omg.org/mda, 2002.
[OMG03] Object Management Group, MDA Guide Version 1.0.1, June 2003
[OMG05] EDOC specification: ftp://ftp.omg.org/pub/docs/ptc/02-02-05.pdf, 2005
[OMG06] OMG: BPMN 1.0.: Business Process Modeling Notation. http://www.bpmn.org/Documents/OMG-02-01.pdf, Feb 2006.
[OMG08] Object Management Group, Software & Systems Process Engineering, Metamodel Specification (SPEM), Version 2.0, April 2008.
[OMG08b] OMG: Busines Motivation Model, http://www.omg.org/spec/BMM/1.0/PDF/, retrieval 2.11.2008
[OMG09] BPMI and OMG: Business Process Modelling Notation Specification. Final Adopted Specification.http://www.bpmn.org/., 2009
[OMG09b] SoaML: Service Oriented Modelling Language V.1.0., beta 2, http://www.omg.org/spec/SoaML/1.0/Beta2/, 2009
[OMG10] Business Motivation Model (BMM): http://www.omg.org/spec/BMM/1.1/PDF/, 1.5.2010
[OMG11] Business Process Modeling Notation. BPMN2.0.: http://www.omg.org/spec/BPMN/2.0/PDF/

[OP10] Osterwalder A., Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. Wiley, 2010
[Ope02]The Open Group: The Open Group Architecture Framework (TOGAF) – 2002
[ORAC11] ORACLE: Oracle Unified Method (OUM) for SOA, ORACLE 2011
[OvdA+06] Ouyang, C., van der Aalst, W., Dumas, M., and ter Hofstede, A. : Translating bpmn to bpel. BPMCenter Report BPM-06-02, BPMcenter.org., 2006
[PM02] Petit M. , Doumeingts G. : UEML : Deliverable D1.1 Report on the State of the Art in Enterprise Modelling, Sept 2002
[Pet07] Pettey C. : 2007 Business and CIO Priorities, http://www.gartner.com/it/page.jsp?id=501189, Feb 2007, Gartner
[Pet08] Peterson G.: Security in SOA, http://www.soamag.com/I15/0208-2.asp, January2008
[Pez06] Pezzini M., Gartner Conference October 2006
[PG07] Pijpers V., Gordijn J.: 'Bridging Business Value Models and Process Models in Aviation Value Webs via Possession Rights', Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS), 2007
[PGA08] Pijpers V., Goordin J., Akkermans H.,: Business Strategy-IT alignment in a multi-actor setting, A mobile e-service case, 2008
[PGAa09] Pijpers V., Gordijn J., Akkermans H.: Exploring inter-organizational alignment with e3alignment ? An Aviation Case. In BLED 2009 Proceedings, Pages cdrom, 2009
[PGAb09] Pijpers V., Gordijn J., Akkermans H.: E3alignment : Exploring Inter-Organizational Alignment in Value Webs. In Proceedings of the Third International Conference on Research Challenges in Information Science, 2009. RCIS, 2009
[PGAc09] Pijpers V., Gordijn J., Akkermans H.: E3alignment: Exploring inter-organizational alignment in networked value constellations. In INTERNATIONAL JOURNAL OF COMPUTER SCIENCE & APPLICATIONS, Vol. VI(V):59-88, 2009
[Pon12] Pondent C.: SOA Architectural Styles and Standards on http://www.ehow.com/list_7620339_soa-architectural-styles-standards.html, Retrieval 6.3.2012
[Por85] Porter M.: Competitive Advantage: Creating and Sustaining superior Performance, 1985
[Pra09] Prado J.: Strategic Business and IT Alignment Assessment: A modeling approach associated with enterprise arquitecture. In: Industrial Information and Control Systems, KTH, p. 48. Royal Institute of Tecnology, Stockholm, 2009
[PRCLM+04] Stanley Presser, Jennifer M. Rothgeb, Mick P. Couper, Judith T. Lessler, Elizabeth Martin, Jean Martin, Eleanor Singer: Methods for Testing and Evaluating Questionnaires, John Wiley & Sons, Inc. 2004
[PT06] Pulier, E.; Taylor, H. : Understanding enterprise SOA. Greenwich, Conn : Manning. 2006
[PTDL06] Papazoglou P.M., Traverso P., Dustdar S., Leymann F.,: Service oriented computing Research Roadmap, March 2006
[PV06]Papazoglou P., van der Heuvel W.J.: Service-oriented Design and Development Methodology, Whitepaper, Int. J. of Web Engineering and technology (IJWET), 2006
[QBP87] Quinn JB., Baruch JJ., Paquette PC.: Technology in services. Scientific American,257 (6), 1987
[Ral99] Ralyté, J.: Reusing Scenario Based Approaches in Requirements Engineering Methods: Crews Method Base. In Proceedings of the 10th Int. Workshop on Database and Expert Systems Applications (DEXA'99), 1st Int. Rep'99 Workshop, Florence, Italy, 1999.
[ResSOA09] Research and Markets. Worldwide SOA component services market shares, strategy, and forecasts, 2009 to 2015 http://www.researchandmarkets.com/reports/838370/worldwide_services_oriented_architecture_soa, 2009

[RGY05] van der Raadt B., Gordijn J., Yu E.: Exploring Web Services from a Business Value Perspective, RE '05 Proceedings of the 13th IEEE International Conference on Requirements Engineering, IEEE Computer Society Washington, 2005
[RM06] Recker J., Mendling J.: On the translation between BPMN and BPEL: Conceptual mismatch between process modeling languages. In 11th Int. Works. on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD'06), pages 521–532, Jun 2006
[Rol08] Roland C.: Method Engineering: Towards Methods as Services. Keynote speech ICSE0. 2008.[RP96] Rolland, C., Prakash, N.: A Proposal for Context-Specific Method Engineering.
In Method Engineering. Principles of Method Construction and Tool Support. Proceedings of IFIP TC8, WG8.1/8.2 Working Conference on Method Engineering, 26-28 August 1996
[RRIG09] Recker J., Roseman M., Indulska M., Green P.: Business Process Modelling – A comparative Analysis , Journal of the Association of Information Systems, Volume 10 Issue 4, 2009
[RS97]: Ross, D.T., Schoman Jr., K.E.: Structured Analysis for Requirements Definition; IEEE Transactions on Software Engineering SE-3, 1997
[RtHE+05] Russell, N., ter Hofstede, A., Edmond, D., and van der Aalst, W. M. : Workflow data patterns: Identification, representation and tool support. In Proc. of the 24th International Conference on Conceptual Modeling (ER 2005).
[RWR06] Ross J, Weill P., Robertson D.: Enterprise Architecture as Strategy, Harvard Business School Press, 2006
[SAP08] SAP: Accelerated Transformation to SOA, 2008
[Sae93] Saeki et al: A metamodel for representing software specification & design methods, in Prakash et al. proceedings of the IFIP WG8.1 Conference on Information Sstem Development Process, Como, 1993
[SJB09] Satzinger J., Jackson R., Burd S.: Systems Analysis and Design in a Changing World, 5th Edition, Course Technology Publishers, 2009
[Sch04] Scheer: Innovation durch Geschäftsprozessmanagement, 2004
[Sch83] Scheer A.W.: Production Control and Information Systems. Methods and Tools for Computer Integrated Manufacturing, 1983
[Sch93] Scheer A.W: Architecture of Integrated Information Systems (ARIS). DIISM, 1993
[Shu06] Shum A. et al: SOA practitioners guide, BEA Systems, Sept 2006, http://dev2dev.bea.com/pub/a/2006/09/soa-practitioners-guide.html
[Sin07] Sinur J. et al, Predicts 2007: Align BPM and SOA Initiatives Now to Increase Chances of Becoming Leader by 2010, Nov 2006
[SMBG07] Spohrer J., Maglio P.P., Bailey J., Gruhl D.: Steps toward a science of service sytems. IEEE Computer 40, pp. 71–77, 2007
[Smi08] Smith R.: A simpler approach to SOA, http://www.informationweek.com/news/software/soa/showArticle.jhtml?articleID=209904293, Retrieval: 19. August 2008
[SMJ96] G. Spur, K. Mertins, and R. Jochem.: Integrated Enterprise Modelling, Beuth Verlag GmbH, Berlin, 1996
[SOA09] SOA Manifesto: 2nd international SOA Symposium Rotterdam, Netherlands, 2009 www.soa-manifesto.org
[SOAKN09]http://www.soa-know-how.de/, Website, 2009
[SRP02] Sharp, H., Rogers, Y., Preece, J., Interaction Design: Beyond Human-Computer Interaction. John Wiley & Sons, Inc. 2002
[SS93] Spewak S, Hill S..:Eenterprise Architecture Planning: Developing a blueprint for Data, Applications and Technology, Wiley, 1993

[Stei08] Stein S. et al: Evaluation of OrVia Framework for Model-Driven SOA Implementations: An Industrial Case Study. In Dumas M., Reichert M., Shan M-C. Proceedings 6th International Conference, BPM2008, 2008
[Stei09] Stein S.: Modelling Method Extension for Service-Oriented Business Process Management, PhD Thesis, 2009
[SUN04] SUN : Assessing your SOA Readiness, Technical Whitepaper, June 2004
[SV05] Stahl T. and Voelter M., Model-Driven Software Development, 2005
[SZ92] Sowa , J., Zachman, J.: Extending and formalizing the framework for information
[Tha01] Thatte, S.: XLANG. Specification, Microsoft Corp., 2001
[Tho07] Thomas O. et al: Serviceorientierte Architekturen: Gestaltung, Konfiguration und Ausführung von Geschäftsprozessen, Institut für Wirtschaftinformatik, Heft 189, Januar 2007
[Tol95] Tolvanen J.P.: Incremental Method Engineering with Modelling Tools: Theoretical Principles and Empirical Evidence, PHD Thesis, University of Jyväskylä, 1995
[Tran09] Tran H.: View-Based and Model-Driven Approach for Process-Driven, Service-Oriented Architectures, Dissertation, TU Vienna, Oct 2009
[TTJ09] Terlouw, J., Terlouw, L., & Jansen, S.: An assessment method for selecting a SOA delivery strategy: determining influencing factors and their value weights. Proceedings of the Busital workshop, Amsterdam, The Netherlands, 2009
[TZD07] Tran H., Zdun U., and Dustdar S.: View-based and Model-driven Approach for Reducing the Development Complexity in Process-Driven SOA. In Intl. Working Conf. on Business Process and Services Computing (BPSC'07), volume 116 of Lecture Notes in Informatics, pages 105–124, sep 2007.
[TZD08] H. Tran, U. Zdun, and S. Dustdar. View-based Integration of Process-driven SOA Models At Various Abstraction Levels. In R.-D. Kutsche and N. Milanovic, editors, Proceedings of First International Workshop on Model-Based Software and Data Integration MBSDI 2008, Berlin, Germany, pages 55-66, CCIS 8, Springer, April, 2008.
[TZD08b] Tran H., Zdun U., Dustdar S.: View-Based Reverse Engineering Approach for Enhancing Model Interoperability and Reusability in Process-Driven SOAs. In 10th International Conference on Software Reuse (ICSR), Lecture Notes in Computer Science 5030, Beijing, China, pages 233-244, Springer, May, 2008.
[Uem03]UEML Project, EU funded 6th Framework Programme, Deliverable D1.1. Part A: Enterprise Modelling State of the Art, 2003
[VCZD91] Vallespir B., Chen D., Zanettin M., Doumeingts G.: Definition of a CIM Architecture within the ESPRIT Project `IMPACS'," in Computer Applications in Production Engineering: Integration Aspects, G.Doumeingts, J.Browne, M.Tomljanovich (Eds), Elsevier, Amsterdam (1991)pp731-738
[vdA97b] van der Aalst W.M.P.: On the verification of interorganizational workows. Computing Science Reports 97/16, Eindhoven University of Technology, 1997.
[vdA97] van der Aalst W. M. P.: Verification of WorkflowNets. In Azema, P. and Balbo, G., editors, Application and Theory of Petri Nets, LNCS 1248, pages 407- 426., 1997
[vdAtH05] van der Aalst W. M. P., ter Hofstede, A.: YAWL: Yet Another Workflow Language. Information Systems, 30(4):245-275, 2005
[vdAtHK+03] van der Aalst W. M. P., ter Hofstede, A. , Kiepuszewski, B., and Barros, A. P. : Workflow Patterns. Distributed and Parallel Databases, 14(1):5{51., 2003
[vdAtHW03] van der Aalst WMP., ter Hofstede A., Weske M.: Business Process Management: A Survey. Business Process Management, 2003
[Ver92] Vernadat F.: CIMOSA - A European development for enterprise integration. Part 2: Enterprise modelling. In Enterprise Integration Modeling (C. Petrie, Ed.). The MIT Press, Cambridge, MA. 1992.
[Ver96] Vernadat F.: Enterprise Modeling and Integration: principles and applications, Chapman & Hall, 1st edition 1996

[VKZ04] Voelter M., Kircher M., Zdun U.: Remoting Patterns. Pattern Series. John Wiley and Sons, 2004.
[VL09] Viering, G.; Legner, C.: State of Realization and Benefits of Service-Oriented Architectures in Germany, Austria, and Switzerland, Schmietendorf, A.; Fiedler, M.; Dumke, R. (Eds.): BSOA 2009: 4. Workshop "Bewertungsaspekte Serviceorientierter Architekturen" der GI Fachgruppe "Software-Messung und -Bewertung", Darmstadt, Germany, 18.11.2009, Shaker, Darmstadt, 2009, pp. 3-16.
[VS06] Voelter M. , Stahl T..: Model-Driven Software Development: Technology, Engineering, Management. Wiley, 2006.
[VZ93] Vernadat, F., Zelm, M. "Advanced modelling approach to CIM systems." In Advances in Factories of the Future, CIM and Robotics (Cotsaftis, M. and F. Vernadat, eds.), Holland. 1993. pp.
[Wal04] Walonick S.: Survival Statistics, Bloomington, 2004
[WB08] van den Weerd I., Brinkkemper S.: Meta-Modeling for Situational Analysis and Design Methods, Handbook of Research on Modern Systems Analysis and Design Technologies and Applications, Information Science Reference, Hershey PA, 2008
[WBFG03] Wieringa, R., Blank, H., Fokkinga, M., Grefen, P: Aligning Application Arc Architecture to the Business Context. In: Conference on Advanced Information System Engineering (CAiSE 03), 209–225 LNCS 2681,Springer 2003
[Wee09] van de Weerd I.: Advancing in Software Product Management: An Incremental Method Engineering Approach , PhD dissertation, University of Utrecht, July 2009
[Weg03] Wegmann A.: On the systemic Enterprise Architecture Method(SEAM), International Conference on Enterprise Information Systems 2003 (ICEIS), Angers, 2003
[WFMC02] Workflow Management Coalition: Workflow Process Definition Interface : XML Process Definition Language. Document Number WFMC-TC-1025, Version 1.0, Workflow Management Coalition, 2002
[WFMC94] The Workflow Management Coalition: Interface 1: Process definition interchange process model. Technical Report WfMC TC-1016-P, July 1998.
[Wie10] Wieringa R.: Design Science Research Methodology: Principles and Practice, RE 10 Tutorial University of Twente, 28.9.2010
[WIKI10b] Wikipedia: List of Web Service Standards, http://en.wikipedia.org/wiki/List_of_Web_service_specifications, April 2009
[Wil92] Williams, T. J., ``The Purdue Enterprise Reference Architecture," Instrument Society of America, Research Triangle Park, NC (1992)
[WKB89] Weisberg, H., Krosnick, J. A., & Bowen, B.: Introduction to Survey Research and Data Analysis, Chicago: Scott, Foresman.,1989, http://stanford.edu/group/polisci/faculty/krosnick.html
[WM06] Woods, D.; Mattern, T. : Enterprise SOA: designing IT for business innovation. 1. Aufl. Beijing O'Reilly, 2006
[YM93] Yu E. and Mylopoulos J.: An actor dependency model of organizational work - with application to business process reengineering. In COCS '93: Proceedings of the conference on Organizational computing systems, pages 258–268, New York, NY, ACM Press, 1993.
[YM96] Yu E., Mylopoulos J. : Using goals, rules, and methods to support reasoning in business process reengineering. International Journal of Intelligent Systems in Accounting, Finance and Management, 5 (1):1-13, January 1996
[Yu95] Yu E. : Models for supporting the redesign of organizational work. In COCS '95: Proceedings of conference on Organizational computing systems, pages 226–236, New York, NY, 1995. ACM Press.
[Yva06] Yvanov K.: ARIS Value Engineering for SOA, IDS Scheer AG, 2006
[Zach87] Zachman, J.: A framework for information systems architecture. IBM Systems Journal 26 26(3), 1987

[ZD06] Zdun, U., Dustdar, S.: Model Driven Integration of Process Driven SOA Models, Whitepaper 2006
[Zdu05] Zdun U.,: Frag: http://frag.sourceforge.net/, 2005
[Zee00] Zee, J.T.M. van der (2000). The BtripleE framework: Measuring the business value of Information Technology. In Proceedings of the 2000 IRMA International Conference on Challenges of Information Technology Management in the 21st Century (pp. 274-278). Hershey, PA: Idea Group Publishing.
[ZGK+07] Zimmermann O, Gschwind T., Kuester J., Leymann F., Schuster N.: Reusable architectural decision models for enterprise application development. In S. Overhage and C. Szyperski, editors, Quality of Software Architecture (QoSA) 2007, Lecture Notes in Computer Science, Boston,
[ZHvdA06] Zdun U., Hentrich C. , v.d.Aalst.: A survey of patterns for service-oriented architectures. International Journal of Internet Protocol Technology, 1(3):132–143, 2006.
[Zim09] Zimmermann O.: An Architectural Decision Modeling Framework for Service-Oriented Architecture Design, PhD Thesis, University of Stuttgart, March 2009.
[ZM05] Ziemann J., Mendling J.: EPC-based modelling of BPEL processes: a pragmatic transformation approach. In Proc. of the 7th Int. Conference "Modern Information Technology in the Innovation Processes of the Industrial Enterprises" (MITIP 2005), 2005.
[zMB99] zur Mühlen, M., Becker, J.: Towards a Classification Framework for Application Granularity in Workflow Management Systems. In: Proceedings of the 11th International Conference on Advanced Information Systems Engineering (CAISE 99). Heidelberg, Germany, 1999. S. 411-416.
[ZMCO04] Zimmermann O., Milinski M., Craes M., Oellermann F.: Second generation web services-oriented architecture in production in the finance industry. In OOPSLA Conference Companion, 2004.
[zMR99] zur Muehlen M., Roseman M.: Evaluation of Workflow Management Systems Using Meta Models. In: Proceedings of the 32nd Hawaii International Conference on System Sciences. Wailea, HI 1999.
[zMue02] zur Muehlen M: Workflow Based Process Controlling, Foundation, design, and application of workflow-driven process information systems, 2002
[zMue99] zur Muehlen M.: Evaluation of Workflow Management Systems using Meta Models, HICSS 1999
[ZZGL08] Zimmermann O., Zdun U., Geschwind T., Leymann F.: Combining Pattern Languages and Reusable Architectural Decision Model into Comprehensible Design Method,. In Working IEEE/IFIP Conference on Software Architecture (WICSA) 2008, Vancouver, BC, Canada, February 2008.

# APPENDICES

## *Appendix A : Questionnaire (Chapter 3.3.)*

PhD Thesis on SOA Methods for process oriented implementation

Please take 30 minutes to fill the online questionnaire. You will benefit from the executive summary published on BPtrends and IT News after having filled the questionnaire. This research is supported by the Luxembourgish Ministry of Education (Ministère de l' Education Nationale et de la Formation Professionnelle) and a close collaboration with Luxembourg's research institute CRP Henri Tudor, Center for IT Innovation (CITI). Should you have any questions or should you have interest in my published articles, please do not hesitate to contact (email to: jan.ricken@fundp.ac.be) me!

**Table 48: Questionnaire Template for section 3.3.**

| 1) Name of your Company / Organization |
|---|
| |

| 2) Please enter your Country: |
|---|
| |

| 3) What is your position within the Company? | |
|---|---|
| Business Analyst | |
| CIO | |
| CEO | |
| Director | |
| Member of IT | |
| Process Manager/Analyst | |
| Other (Please Specify): | |
| | |

| 4) Please enter your e-mail address. It is required to send you the promised executive summary after research analysis. |
|---|
| |

| 5) What is your industry you are working in? | |
|---|---|
| Agriculture and food industry | |
| Banking and Finance | |
| Beverages and tobacco | |
| Communication and telecommunication | |
| Consulting & Audit | |

| | |
|---|---|
| Construction and finishing | |
| Chemical and phama | |
| Data processing | |
| Electric and electronic industry | |
| Energy | |
| Insurance | |
| Metal and Steel | |
| Public Sector | |
| Printing and paper converting | |
| Steel industry | |
| Services | |
| Temporary work | |
| Transport,handling & Logistics | |
| Various industrial services | |
| Waste management and transport | |
| Automotive | |
| Other (Please Specify): | |
| | |

| 6) How many staff do you employ (status beginning 2008)? | |
|---|---|
| <50 | |
| 50-100 | |
| 101-500 | |
| 501-1000 | |
| 1.001-1.500 | |
| 1.501-10.000 | |
| >10.000 | |

| 7) If applicable, what is the approx. turnover in EUR of your company recently? |
|---|
| |

| 8) Do you answer the questionnaire for your Headquarter or a Branch/Subsidiary? | |
|---|---|
| Headquarter | |
| Branch/Subsidiary | |

| 9) How many applications/software systems do you manage? | |
|---|---|
| <10 | |

| | |
|---|---|
| 11-25 | |
| 26-50 | |
| 51-100 | |
| 101-500 | |
| >500 | |

| 10) Do you know the concept of SOA? If yes, since when (pls. indicate year)? | |
|---|---|
| Yes | |
| No | |
| Other: | |
| | |

| 11) Is SOA a paradigm you want to use? | |
|---|---|
| Yes | |
| No | |

| 12) If yes, how far are you? | |
|---|---|
| Planned | |
| Project is ongoing | |
| already implemented | |
| Other (Please Specify): | |
| | |

| 13) What is following your expectations the biggest advantage of SOA? (Put into a ranking) | |
|---|---|
| Reduction of IT cost | _____ |
| Flexibility and Agility in IT Architecture by re-using services | _____ |
| Business and IT Alignment by common views and language | _____ |
| Enforcement to think in processes | _____ |
| Re-utilisation of existing BPM content | _____ |
| Automatically enforces data quality / data management | _____ |

| 14) What are following your expectations the biggest challenges of SOA? (Put into a ranking) | |
|---|---|
| ROI difficult to calculate | _____ |
| Tangible benefits hard to identify | _____ |
| Complexity of subject | _____ |
| Knowledge & right profiles | _____ |
| Missing approach and where to start | _____ |

| | |
|---|---|
| Organizational Alignment | _____ |
| Change Management | _____ |
| SOA Governance (Roles&Responsibilities) | _____ |
| Top Management Buy-in | _____ |

| 15) Enterprise Architecture and Frameworks | | | | | |
|---|---|---|---|---|---|
| | Not Known | Known | Known & used, Meeting Expectations | Known & used, Not Meeting Expectations | Other (Please Specify): |
| GRAAL framework | | | | | |
| The Zachman Framework | | | | | |
| The Four-Domain-Architecture | | | | | |
| TOGAF | | | | | |
| RM-ODPDoDAF/C4ISR | | | | | |
| GERAM Generic Enterprise Reference Architecture and Methodology | | | | | |
| Nolan Norton Framework | | | | | |
| CEN ENV 40 003 | | | | | |
| CIMOSA | | | | | |
| GRAI/GIM | | | | | |
| PERA | | | | | |
| ARIS | | | | | |
| TOVE | | | | | |
| 4+1 View Model of Architecture | | | | | |
| Model Driven Architecture (MDA) | | | | | |
| RUP: Rational Unified Process | | | | | |
| ArchiMate | | | | | |
| TEAF | | | | | |
| AKM | | | | | |

| 16) Modelling Languages & Model Types | | | | | |
|---|---|---|---|---|---|
| | Not Known | Known | Known & used, Meeting Expectations | Known & used, Not Meeting Expectations | Other (Please Specify): |
| ARIS | | | | | |
| ArchiMate | | | | | |
| BSC | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| BPEL | | | | | |
| BPDM | | | | | |
| BPML | | | | | |
| BPMN | | | | | |
| BOP | | | | | |
| CIMOSA | | | | | |
| CORBA IDL | | | | | |
| e3-Value | | | | | |
| ebXML | | | | | |
| EEML | | | | | |
| EKS | | | | | |
| EPC | | | | | |
| EDOC | | | | | |
| GRAI/GIM | | | | | |
| IDEF | | | | | |
| IEM / MO2GO | | | | | |
| jPDL | | | | | |
| MEMO | | | | | |
| METIS Enterp | | | | | |
| MOF | | | | | |
| MEML | | | | | |
| Petri Nets | | | | | |
| PIM4SOA | | | | | |
| PIF | | | | | |
| PSL CORE | | | | | |
| SADT | | | | | |
| SPEM | | | | | |
| Testbed | | | | | |
| UEML | | | | | |
| UML | | | | | |
| Value Chain | | | | | |
| WSDL | | | | | |
| WPDL | | | | | |
| XPDL | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| YAWL | | | | | |

| 17) What approach do you have chosen for modelling and implementing SOA? | |
|---|---|
| top-down | |
| meet-in-the-middle | |
| Bottom-up | |
| Other (Please Specify): | |
| | |

| 18) Our Company is using a management method (e.g. Balanced Score Card, Management Cockpit etc.) to derive from business strategy the process objectives and IT objectives... | |
|---|---|
| Yes | _____ |
| No | _____ |

19) Principles of Model Driven Architecture (MDA) from the OMG are...

| | Not Known | Known | Known & used, Meeting Expectations | Known & used, Not Meeting Expectations | Other (Please Specify): |
|---|---|---|---|---|---|
| Software Development | | | | | |
| SOA Implementation | | | | | |

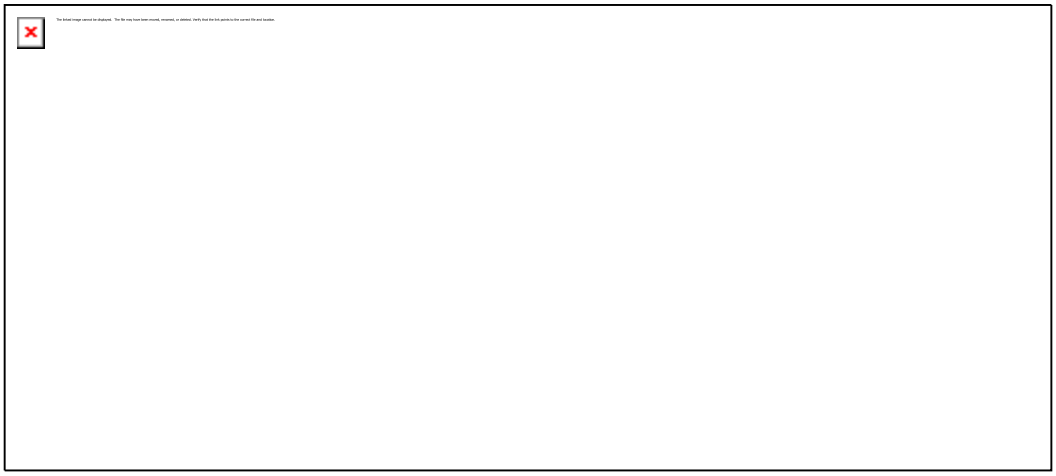
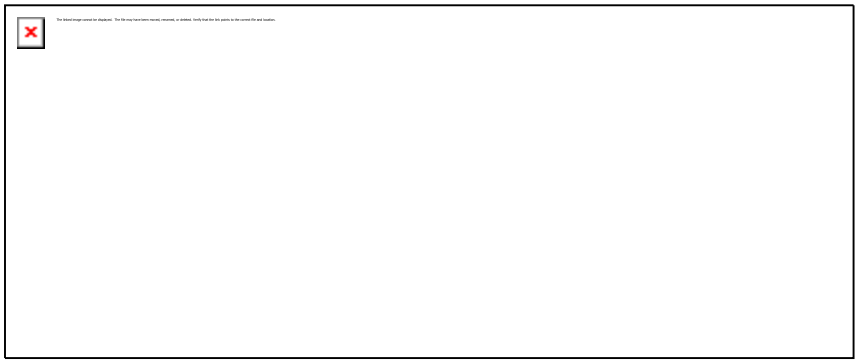


Figure related to MDA abstraction layers

| 20) Question related to above Figure: What type of models do you use for the different abstraction levels? | |
|---|---|
| Platform-independent level (PIM) | |
| Computer-Independent level (CIM) | |
| Platform specific level (PSM) | |

21) Do you transform automatically technical models e.g. UML into Software Code or service descriptions?

| | |
|---|---|
| Yes | _____ |
| No | _____ |

22) 

| | used | used & successful | used & failure | n/a | Other (Please Specify): |
|---|---|---|---|---|---|
| e3value2ADM | | | | | |
| ADM2BPMN | | | | | |
| EPC2BPMN | | | | | |
| EPC2UML | | | | | |
| EPC2BPEL | | | | | |
| BPMN2BPEL | | | | | |
| UML2BPEL | | | | | |
| BPEL2WSDL | | | | | |
| UML2WSDL | | | | | |

| 23) Do you manage our processes through a Business Process Management (BPM) – programme (e.g. Strategy, Design, Implementation, Controlling, Change Mgt.)? | |
|---|---|
| Yes | |
| No | |
| Partly | |

24) Which of the following BPM usage Scenarios do you have?

| | Yes | No | Planned | Other (Please Specify): |
|---|---|---|---|---|
| Documentation | | | | |
| Certification | | | | |
| Risk Mgt | | | | |
| Cost Improvement | | | | |
| Process-Driven Application Development | | | | |
| Process-Driven Web Service Construction | | | | |

| 25) How do you rate the importance of BPM knowledge for SOA implementation? | |
|---|---|
| Very important | |
| neutral | |
| not important | |

| 26) Do you use SOA Maturity model? | |
|---|---|
| Yes | _____ |
| No | _____ |

| 27) If your answer above is YES, which maturity model do you use? |
|---|
| |

| 28) Did you succeed to calculate ROI for your SOA project? | |
|---|---|
| No, We did not succeed | |
| Yes, ROI 1-2y, | |
| Yes, ROI 2-3y, | |
| Yes, ROI 3-5y, | |
| Yes, ROI >5y | |

| 29) Does your company has.... | | | | |
|---|---|---|---|---|
| | Yes | No | Not applicable | Other (Please Specify): |
| A strong business case for SOA? | | | | |
| Tendentially more business skills? | | | | |
| Tendentially more technical skills? | | | | |
| The right skills to understand SOA? | | | | |
| The right skills to implement SOA? | | | | |
| The need to get external help? | | | | |

| 30) We use the following project management method for all (IT) projects | |
|---|---|
| PMI | |
| PRINCE2 | |
| SUMMIT | |
| Own Methodology | |
| Other (Please Specify): | |
| | |

| 31) For SOA specifically, please rate the following SOA Methods (alphabetical order): |
|---|

| | Not Known | Known | Known & used, Meeting Expectations | Known & used, Not Meeting Expectations | Remark / Please Specify: |
|---|---|---|---|---|---|
| ARIS Value Engineering for SOA, (IDS Scheer AG, 2006) | | | | | |
| Enterprise SOA, (SAP AG, 2006) | | | | | |
| Enterprise SOA Adoption Strategies, (Capgemini 2006) | | | | | |
| Model-Driven Integration of Process driven SOA Models, (Distributed Systems Group, 2006) | | | | | |
| Platform-independent model for service-oriented architecture, (European Software Institute (ESI) Spain, DFKI GmbH Germany, SINTEF ICT, Norway) | | | | | |
| Service-oriented Design and Development Method(Papazoglou, University of Tilburg. June 2007) | | | | | |
| Service oriented Modelling & Architecture, (IBM, 2006) | | | | | |
| Oracle Unified Method for SOA | | | | | |
| Other Methodology | | | | | |

| 32) In general, SOA method is very complex and not trivial to tackle... | |
|---|---|
| True | _____ |
| False | _____ |

| 33) Do you have written SOA Objectives, Key Performance Indicators, SOA Drivers & Critical Success Factors identified? | |
|---|---|
| Yes | |
| No | |
| Partly | |
| Planned | |
| Other (Please Specify): | |
| | |

34) What Tools BPM/SOA Design & BPM/SOA Runtime Tools do you know? What is your experience?

| | Not Known | Known | Known & used, Meeting Expectations | Known & used, Not Meeting Expectations | Remark/ Please Specify: |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| ARIS | | | | | |
| Casewise | | | | | |
| Intalio | | | | | |
| MEGA | | | | | |
| Metis | | | | | |
| Nautilus | | | | | |
| Popkin | | | | | |
| Visio | | | | | |
| Other BPM Design Tool | | | | | |
| BEA | | | | | |
| HP | | | | | |
| IBM | | | | | |
| MICROSOFT | | | | | |
| ORACLE | | | | | |
| SAP | | | | | |
| SUN | | | | | |
| Other BPM Run-time Tool | | | | | |

| 35) Questions related to web services.... | | | | |
|---|---|---|---|---|
| | Yes | No | Partly | Remark/Please Specify: |
| Service orientation is part of our business strategy | | | | |
| Our IT is outsourced or partly outsourced | | | | |
| We use already service technology | | | | |
| We start from the services and develop step by step interesting new services for the business | | | | |
| The business is asking IT to develop services to perform better implemented business processes | | | | |
| We deploy services to other organizations and measure by SLA | | | | |
| We have a SOA security management (Authentication, Authorization, Identity Mgt.) | | | | |
| Decomposition of Web Services and Quality of Web-Services are still big challenges | | | | |
| We manage our data through data | | | | |

| | | | | |
|---|---|---|---|---|
| management programme | | | | |
| We master the interfaces between systems | | | | |
| Our Application Interfaces are automated | | | | |
| | | | | |

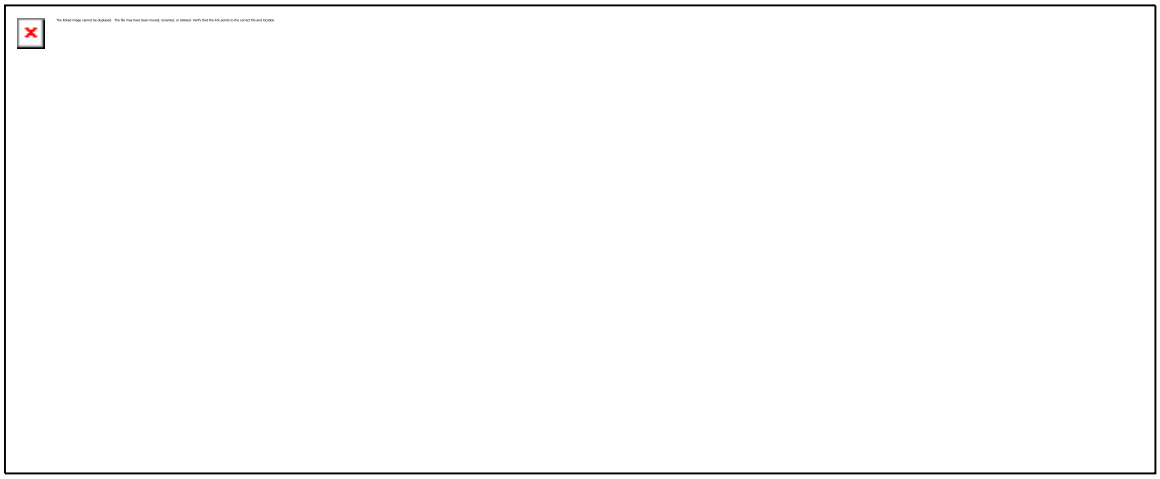


Figure: SOA Method Domain Model

| 36) The presented model is reflecting all domains to consider for an exhaustive SOA implementation method based on a process-oriented approach... | |
|---|---|
| I agree | |
| I do not agree, this is missing: | |
| | |

## *Appendix B: Content of SOA Methods*

Each method is presented, discussed and structured following to specific criteria's:

**Source:** Commercial Organisation/Software Vendor, Independent Authors of books, Academic Researchers

**Viewpoint:** Mostly, the background of the authors is determining if the method is technical, functional or equilibrated

**Approach**: Literature differentiates Top-down, Meet-in-the-middle, and Bottom-up

The chapter is outlining the content; the summary gives a detailed neutral explanation of the methodology, whereas the comment explains the strengths and weaknesses of the methodology.

### Architecting Industry Standards for Service Orientation

**Principal Author:** J. Lee [Lee05]
**Company/Organization:** Microsoft
**Year of Release:** 2005
**Category:** Whitepaper
**Nb. of pages:** 14
**Source:** Commercial Organisation/Software Vendor
**Viewpoint:** technical (PSM)
**Approach**: no approach
Web: http://msdn2.microsoft.com/en-us/library/ms978270(d=printer).aspx

**Chapters:**

1. Introduction
2. Service Orientation Basics
3. Standard Message Composition
4. Headers Are for Standards Too
5. Achieving Interoperability
6. Best Practice Summary
7. More Work to Do
8. Conclusion

**Summary:**

After a very brief introduction (187 words), the author describes in the next chapter "Service Orientation Basics" the four core tenets:

- Service boundaries are explicit.
- Services are autonomous.

- Services share schema and contract, not class.
- Service compatibility is determined based on policy.

The next chapter “Standard Message Composition” lists related to experience in industry four key standards architectures:

- Large and Bulky.
- Service Message Grouping.
- Message Granularity.
- Bits and Pieces.

The four categories are described and the concept of service orientation is introduced. An example of a poorly defined web-service in WSDL is given and compared to an accurately factored message. The difference between both examples is explained in detail. Schemas and how messages are technically decomposed is explained.

The next chapters are used to explain about web service policy, integrity, security and message versioning. The two levels of web-service interoperability are explained. The best practice summary focus on how web-services should be designed:

- Compose granular messages. Use a data dictionary to build discreet and granular messages that will leverage a namespace to align the data payload to the service and data.
- Avoid payload bloat.
- Create service-to-message correlation.
- Use strong naming techniques. Use <import> of global types.
- Avoid schema bloat.
- Support industry standards.
- Use WS-Policy statements to enforce compatibility. Support the XML Schema discovery and Web Service Proxy Model.
- Follow interoperability guidelines for services.
- Support a mainstream Web services stack.

The author concludes by summarizing and highlighting the importance of well-designed web-services.

**Comments:**

This whitepaper is targeting technical specialist with responsibilities to design web-services. However, the content is very technical language. It cannot be considered as a method for the implementation of SOA, but more as a whitepaper for web-service developers.

## ARIS Value Engineering for SOA

**Principal Author:** K. Ivanov [Yva06]
**Company/Organization:** IDS Scheer AG
**Year of Release:** 2006
**Nb. of pages:** 45
**Category**: Presentation of ARIS value engineering for SOA, ARIS Process World 2007 Berlin
**Web**: -
**Source:** Commercial Organisation/Software Vendor
**Viewpoint:** functional & technical (CIM-PIM-PSM)
**Approach**: Top-down

**Chapters:**

1.) Why companies need SOA
2.) SOA – what is behind?
3.) Business Driven SOA
4.) ARIS solution for business-driven SOA
5.) SOA implementation
6.) Best practice examples

**Summary:**

Overall, the method is structured into 4 phases: Strategy, Design, Implementation, and Controlling

The first chapter clarifies about strategic positioning and the related strategic objectives. General common objectives from CEO, CIO, and CFO are explained.

The second chapter tells in brief what SOA is and distinguishes business goals and IT goals. Business goals such as

- Enabling fast production of new business models
- Attaining adaptability to support on-going change
- Accomplishing a closer alignment of IT with business needs
- Achieving higher productivity of Business Processes

IT Goals such as

- Enabling greater re-use of IT assets
- Reducing development cost and project times
- Achieving faster delivery of value to the business
- Accomplishing a higher degree of effectiveness in implementation, modification, and integration of IT systems.

Therefore, processes answer SOA questions e.g. the identification of services, impact of services to business etc.

The authors define a business BPM (Business Process Definition, Rule definition, Business Services and Data Definition, Enterprise Architecture) and a technical BPM (Business Rule Execution, Software Development, Process Execution, Service Implementation & Deployment). Between both, an integration layer (Software Architecture (UML), Service Orchestration (BPEL), Service Design (WSDL) and Business Rule Transformation) interfaces both levels. The author positions the approach on the first two levels, whereas other commercial vendors cover level 2 and 3. (See picture) Furthermore, roles are identified with activities that should be performed on the different levels.

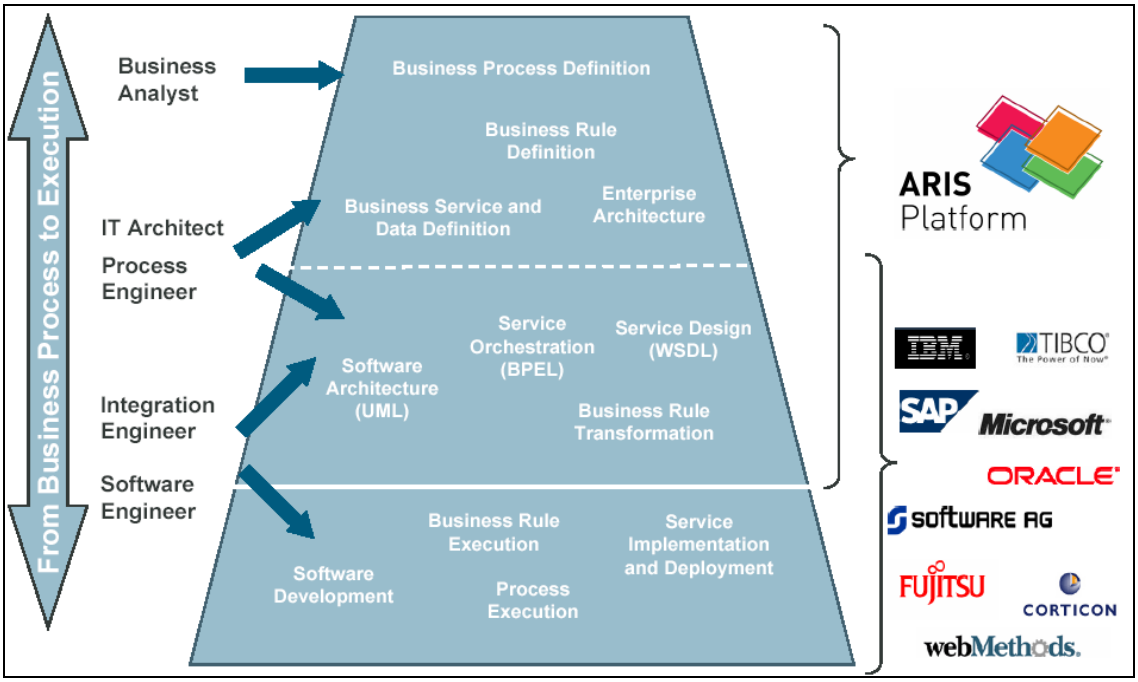



**Figure 80: Levelling of Design Time Tool vs. Run Time Tool**

The solution scenarios that can be covered by the method are three-fold:

1.) Service Architecture Management: Enabling consistent business-driven service architecture to be created for all organizational units and implemented in SOA projects for company-wide re-use.
2.) Service Orchestration & Process Automation: Building of high-value business services orchestrations as input for process execution engines using business and service architecture
3.) Service & Application Engineering: Development of services and applications based on business requirements using UML based object-oriented analysis and design.

The ARIS AVE method differentiates conceptually the SOA design time and the SOA run time:

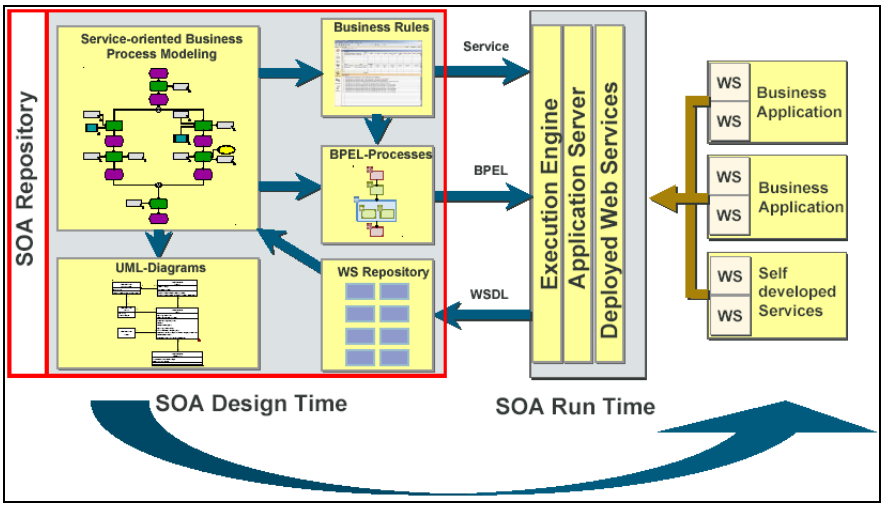



**Figure 81: IDS Scheer link between SOA Design Time and SOA Run Time**

The process to service transformation is done as follows:

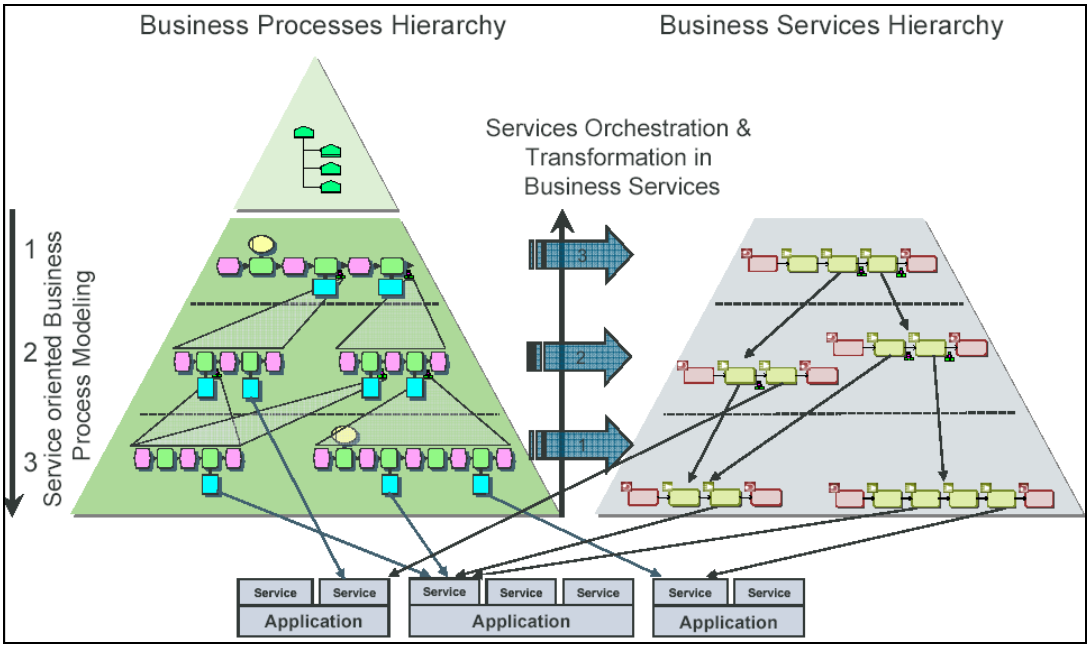



**Figure 82: Process Service Transformation**

All the steps of the SOA roadmap per phase can be seen in the following picture:

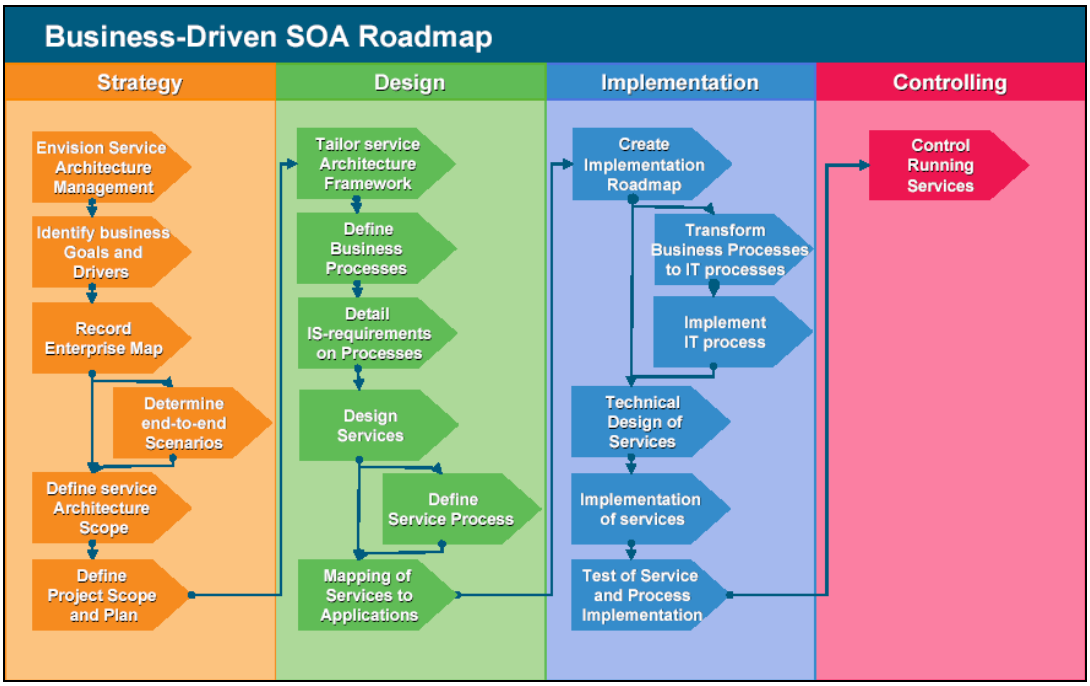



**Figure 83: Business Driven SOA Roadmap by IDS Scheer**

The next chapter shows the service architecture repository and the links between processes, services, systems, and components. (See picture)

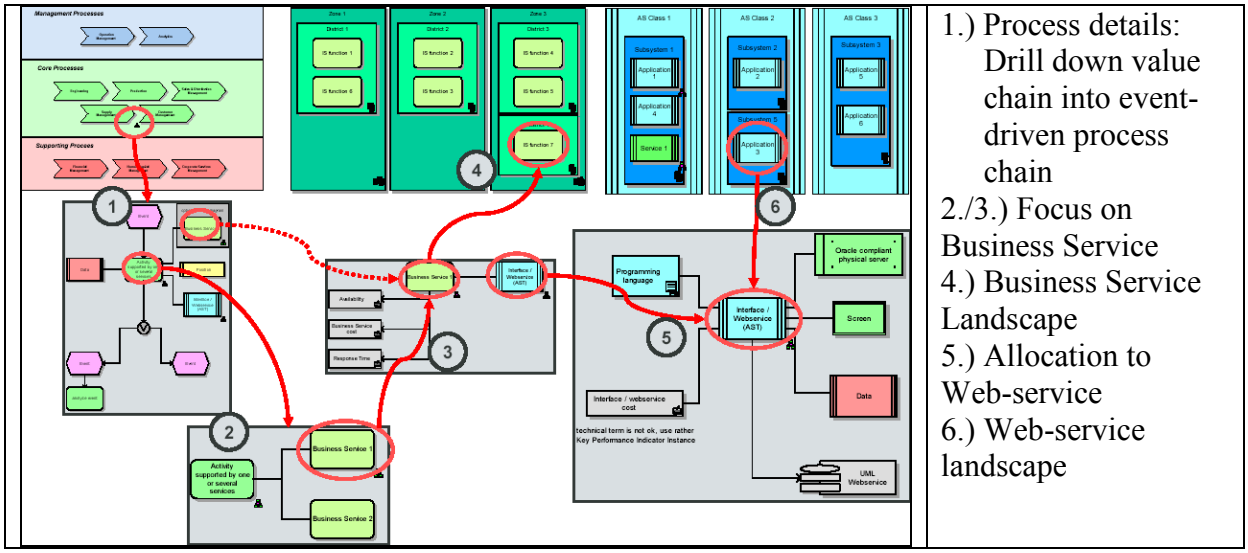



**Figure 84: Modeling links of IDS Scheer approach**

Furthermore, the method foresees an upload of available WSDL services and the link of those services within UML models. The services are embedded into process logic, including rules and events, and can then from the process model (Event-driven-process-chain) automatically translated into Service Orchestration Models in BPEL models/language. This is done on the above mentioned level 2, where the integration interface starts to the technical SOA or run-time environment. BPEL models can then be implemented and executed in tools such as IBM Websphere, SAP Integration Builder, or ORACLE JDeveloper.

**Comments:**

The method is well structured into 4 main phases, beginning with strategy. Here is one of the main strengths of the method, because the strategy effect is related to a consistent top-down method. Only business relevant strategies, objectives, critical success factors, and scoping are the starting point for questions that could be resolved by web service enabled IT structures. It is well explained, what different models can be used on each level but not very exhaustive. Also worth mentioning the strictly functional approach based on modelling that IDS Scheer positions itself on the functional or the so called "SOA design time" against the other big commercial vendors as "SOA run time" (SAP, IBM, Microsoft, ORACLE, BEA etc.) with their technical implementation solutions. However, MDA is not linked to the levels, but the available method and models can be mapped to MDA. Beside the method, IDS Scheer is using BPM tools allowing designing business requirements in a controlled and integrated way. The method could be enlarged by subjects explained more in detail in other methods such as governance, QoS, Web service granularity, technical environments, service decomposition, Master Data Management etc.

## Assessing your SOA readiness

**Principal Author:** -
**Company/Organization:** SUN Mircosystems, [SUN04]
**Year of Release:** 2004
**Category:** Whitepaper
**Nb. of pages:** 9
**Web:**
**Source:** Commercial Organisation/Software Endor
**Viewpoint:** functional
**Approach**: Top-down

**Chapters:**

1. Overview
2. What is SOA?
3. The Benefits of SOA
4. Challenges in Moving to SOA
5. SOA Impact Analysis
6. Technology and Tools
7. Organizational Alignment
8. Method and Process
9. Recommended Approach
10. SUN´s SOA Readiness Assessment
11. Additional SUN SOA Service Offerings
12. Getting Started

**Summary:**

After a quick introduction and a brief clarification, what SUN understands about SOA, the potential benefits are listed and challenges are described. SUN structures the explanation in design-time-environment and run-time environment. Exchange patterns are considered as critical and more success factors are explained: Identity, Registration/Discovery, Service API, Tiering/Layering, Loose Coupling, Pattern Usage, Creation&Deployment, Standardized Data Model, Separation of Business and IT Services, Interoperability and Open Standards. Next, the organizational alignment strategy critical success factors are: Shared Service Strategy and Funding Model. Furthermore in the section "method and process" the author focus on Governance Model and Model-Driven-Architecture. The recommended SUN method is based on 4 steps: Education, Assessment, Planning and Execution. The last three chapters are dedicated to the SUN service offer related to SOA implementation: a readiness assessment is proposed to identify context, maturity and opportunities.

**Comments:**

The paper gives a short, well-structured introduction in SOA, the challenges and critical success factors. The paper is written in a business/functional language and is easy to understand. The chapters "method and process" and "Recommended Approach" are related to the other chapters too short. Nevertheless, the paper gives a brief first introduction into the subject by focussing on the main areas of interest. The paper gives ideas of things to take into consideration, but it is not going into details how to do so e.g. what models to use, how a technical set-up can be made etc. The target audience of this paper are CIO's, Enterprise Architects or divisional IT representatives with the objective to provide a first introduction into the subject.

## Enterprise SOA: Designing IT for Business Innovation

**Principal Author:** Dan Woods, David Mattern, [WM06]
**Company/Organization**: SAP AG
**Year of Release:** 2006
**Category**: Book
**Nb. of pages**: 423
**Web:**
**Source:** Commercial Organisation/Software Vendor
**Viewpoint:** functional
**Approach**: Top-down

**Chapters:**

1.) ESA in the World of Information Technology
2.) The business Case for ESA
3.) Evolving Toward ESA
4.) ESA fundamentals: Learning to think ESA
5.) The structure of ESA

6.) Enterprise Service Community
7.) Creating a Roadmap with the ESA Adoption Program
8.) The enterprise Service Repository and the Enterprise Service Inventory
9.) Project Mendocino: A product based on Consuming Enterprise Services
10.) ESA at Work: Examples from the field
11.) SAP xApps Composite Applications for Analytics The Architecture and Development Tools of Composite Applications
12.) The Architecture and Development Tools of Composite Applications
13.) Supporting Composite Applications
14.) Web Service Basics
15.) Creating Enterprise Services in ABAP
16.) Creating and Consuming Services in JAVA
17.) ESA and IT Governance
18.) ESA Lifecycle Management and Operations
19.) ESA Security
20.) Standards and ESA

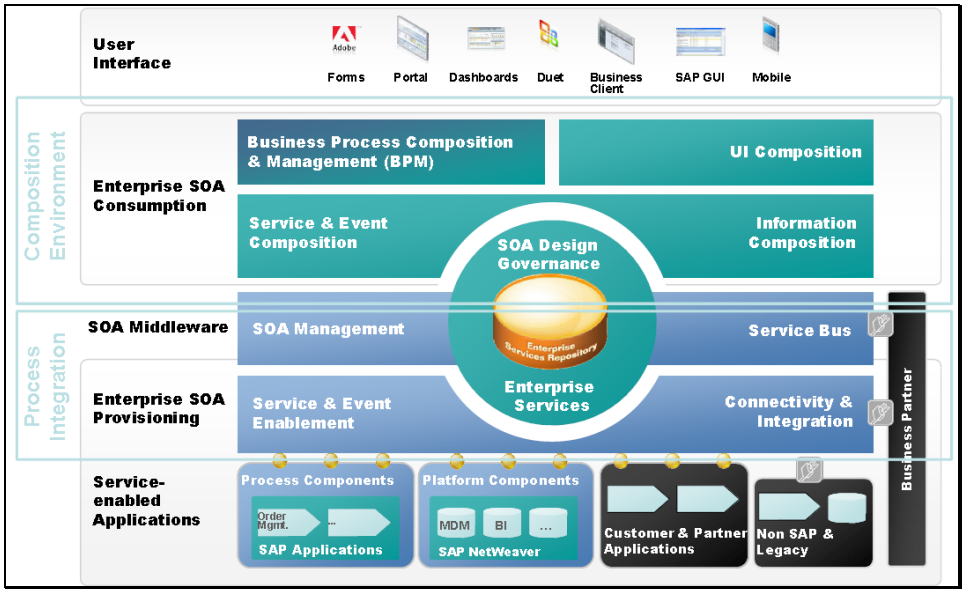



**Figure 85: SAP SOA ESA Overview**

## Summary:

The entire book is about ESA (Enterprise Services Architecture) in relation to the Enterprise Resource Planning (ERP) System SAP from SAP AG.

The first chapter is positioning the book: Audience, challenges, why ESA, web service history, ESA supporting Infrastructure, ESA objectives and benefits and ESA business case. Next the steps for evolving toward ESA are identified and explained: First big obstacle is the enterprise culture and organization that needs to be changed or adapted to new concepts. The role of IT is explained in detail and what new roles and skills are required. Governance in ESA context is roughly explained and the question of modelling interoperability is raised. They state that no standards body or language has so far been recognized as de-facto standard. However, SAP is working with different industry leaders to develop standards. The next chapter "ESA fundamentals" explains again ESA infrastructure, ESA challenges, web-services. The authors differentiate web services and enterprise services and put composite

applications into the context of service oriented architecture. The SAP NetWeaver technology solution map is explained and the concept of event-driven architecture is introduced. Modelling is seen as an important part of ESA: low-specification models and high specification models, pattern based models and requirement models are explained in one sentence. Every ESA Stack is explained layer by layer: User interface, Process orchestration, enterprise service, business objects and persistence.

The chapter of "ESA community" is describing the programme of SAP to bring together partners and customers to share ideas, innovations and web services. Then the chapter "Creating a Roadmap with ESA Adoption Program" presents the method of ESA adoption: Discover, Evaluate, Implement, Operate. Each phase is explained in detail and practical examples from projects are given. The authors refer to SAP methodology. It is said what to do, but not how to do it and by whom. Three case studies with the application of the before explained methodology, Manchette Publicité, Wacker Chemie AG and LHI Leasing, are helping to understand how SAP applies the method.

The chapter "The Enterprise Service Repository and the Enterprise Service Inventory" explains from the ESA viewpoint the utility of the Enterprise Services Repository. One of the fundamental principles of ESA is the business processes as starting points for the design of strategic services that will support those processes. ARIS is a tool to design processes and services is available as separate product, but ESA integration is foreseen in the future. The Enterprise Service Repository based on SAP XI technology is explained (Process Models, Integration Objects, Service and Business Objects). A detailed top-down method and procedural model to define services is explained (p205-211) and a concrete example of the process "purchasing a new component" is given (p. 212-215). "Project Mendocino" is explained: The aim is to integrate desktop applications like Outlook, Excel and Word into SAP tools. Time management of projects through Outlook calendar, budget monitoring, leave management and organization management can be organized more efficiently as processes with related data (times, budget, cost etc.) can be automated. The next chapters are dedicated to composite applications and available development tools (SAP NetWeaver Visual composer, Guided procedures design time for modelling user-centric composite processes, the SAP composite application framework, ABAP Development Workbench and SAP NetWeaver Development Studio). The authors focus on data and especially on master data as a key element to consider. The SAP Master Data Management is explained.

The chapter "Web Service Basics" is introducing a definition for services, SOA, XML, XML Schema, SOAP, WSDL, UDDI followed by a chapter with detailed explanation how to create Web-services / enterprise services with ABAP tools and JAVA tools.

The chapter "ESA and IT Governance" gives an overview about the history, objectives and challenges of managing services in the ESA environment. The last chapters talk about ESA life-cycle (Implementation, Operation, Change Mgt./Continuous Improvement), ESA Security and ESA Standards.

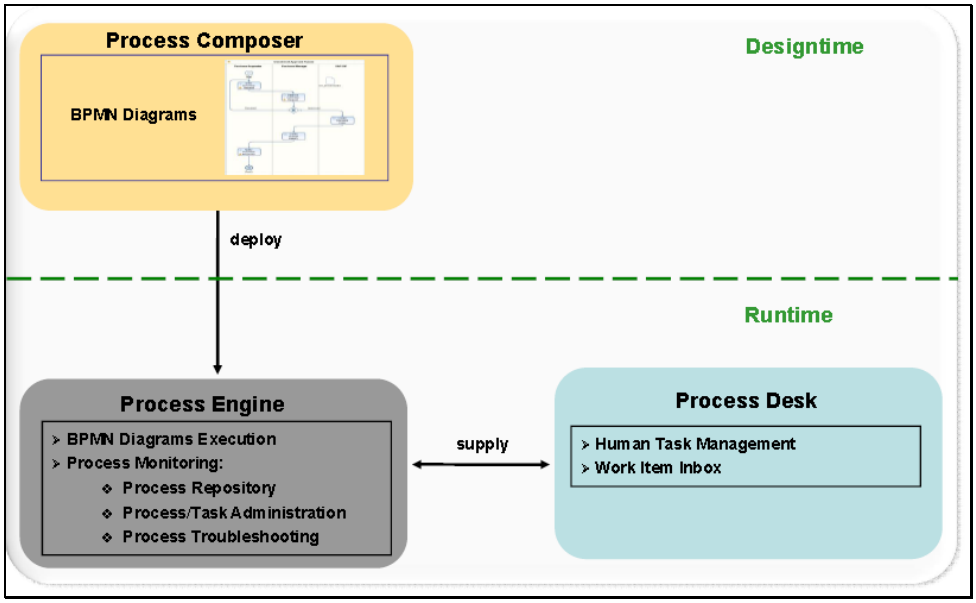



**Figure 86: deployment of BPMN Diagrams into Process Engine**

## Comments:

The authors describe a possible SOA scenario from the perspective using SAP ERP systems in the latest version with XI technology and SAP NetWeaver. The book spans a lot of subjects – but the focus is clearly on the integration of SOA paradigm into the ESA environment. Some explanations in chapters are redundant and definitions are sometimes not very detailed. Some subjects are spread over different chapters instead of focussing into one chapter. (E.g. governance p.83, 379 or web-services, p.24, 99, process orchestration p. 19, 117, 150) Concepts are introduced but only deeply explained in a later chapter. From a didactical point of view, this could be improved. The book mostly describes what needs to be done, but not how it should be done. A pleasant exception is chapter 8 where the approach of creating Enterprise services also explains how it should be done. Sometimes, content is related to SAP system descriptions with rather limited added value. It will be interesting to see if the concept of SAP´s "Enterprise Services Community" will be accepted by companies. The question of ROI for instance is not answered, it is not clear what types of model types the authors recommend on which level and who should do what exactly. The case studies are rather high-level than detailed – again the added value is limited.

Overall, this book is written in functional/business language with target audience CIO, Enterprise Architects, Business Analysts, Senior Executives. It is also only relevant and useful for audience in the context of used SAP ERP and SAP XI / NetWeaver - technology.

The method described in chapter 8 is for sure not complete, but some elements could well be considered for a later definition of an own method.

## A Method for Service Architectures

**Principal Author**: S. Jones, [Jon05]
**Company/Organization:** Capgemini
**Year of Release:** 2005 (Oct)
**Category:** Commercial Whitepaper
**Web**:http://glintech.com/downloads/A%20Methodology%20for%20Service%20Architectures.pdf
**Nb of pages**: 31
**Source:** Commercial Consulting Organisation
**Viewpoint:** functional
**Approach**: Top-down

**Chapters**

1.) Context
2.) Executive Summary
3.) Abstract
4.) Introduction
5.) Overview
6.) A common starting point
7.) Start at the Top
8.) Terminology
9.) Collaborative Working
10.) Creating a Service Architecture
11.) Developing the complete architecture
12.) Managing Change
13.) Summary

**Summary:**

The authors directly from the beginning of the first chapters say that the objective of their proposed method is a top down methodology, based on business visions and not on new technology concepts. The key to SOA is clearly the services. This method does not focus on how to deliver software projects, but to provide the architecture to ensure that the delivery is service oriented. The authors say in the "Overview" that they will cover

- Why services need to be defined
- The importance of a common language
- How to discover what are the primary business services
- How to identify shared and supporting services
- How to define the interactions between services at a high level and
- How to categorize services to help with management.

Clearly excluded is:

- Defining how processes work between services
- Full enterprise or solution Architecture
- The technical requirements of services

- The functional requirement of services
- The implementation of services
- Management of service programmes

The presented method follows the 4 steps:

- What: defining the scope of services,
- Who: Who are the external actors that drive the services or with which the services interact
- Why: Identifying, why one service talks to another
- How: the detail about the processes that co-ordinate the services and also the detail on how a service itself will be implemented

The method starts by the identification of the different divisions/departments their actors and their primary tasks. Once this is done, the interaction between external actors (customer, Logistics Company, suppliers etc.) and divisions/functions is drafted. Then the authors describe a drilling down to level 1 where divisions are split into areas and links and relations are drafted. Virtual Services, Support services and Shared Services are identified. To conduct such a project, the authors describe in chapter 11 the different roles with their responsibilities. The final result should be a big picture showing all services within divisions and sub areas.

**Comments**

The approach is a pure consulting methodology, independently of any tool, environment or standard. It describes step by step the process of the identification of services in a company. The authors assume strictly a top-down approach as the only way to define from business requirements different types of services. Of course, a lot of areas are not in scope, but for the small part that is in scope, it might be a reasonable way to identify services. Every technical aspect is out of scope. It is written in business language and targets CIO, Business Analyst and Enterprise Architects.

## Model-Driven Integration of Process driven SOA Models

**Principal Authors:** Uwe Zdun and Schahram Dustdar, [ZD06], [ZD08], [Tra10]
Company/Organization: Distributed Systems Group, Information System Institute Austria
**Year of Release:** 2006
**Category:** Academic Whitepaper
**Web**: http://drops.dagstuhl.de/opus/volltexte/2006/820/pdf/06291.ZdunUwe.Paper.820.pdf
**Nb. of pages**: 32
**Source:** Academic Organisation (University)
**Viewpoint:** technical (PIM-PSM)
**Approach**: Top-down

**Chapters**

1.) Introduction

2.) Background on MDSD
3.) Model Driven Tool Chain
4.) Meta Model for SOA Integration
5.) DSLS for Flow Models
6.) DSL For Architectural Models
7.) SOA Model Integration
8.) Related Work and Evaluation
9.) Conclusion

**Summary:**

The first chapter gives a quick introduction into SOA and describes the central challenge, which is the integration of different kinds of models and abstractions. Different languages and tools exist with highly different characteristics. It is said that meta-models on the domain-specific languages (DSL) level resolve the issue identified. The approach is based on patterns whereas UML2 and OCL are used to develop a formal modelling language.

The second chapter introduces in detail the used DSL met model and the used UML2 models: Activity diagrams to model flow abstractions and class/component diagrams to model object oriented design and architecture models. The ultimate goal of all transformation consists of generating code in executable language or programming languages.

The third chapter states that UML is the only language that can be considered as a real standard. Related to tools, it is crucial that they support meta models and adapters enable interoperability through code generation. A syntax is describing how the DSL meta model is mapped to language elements and grammar. The authors use Frag textual syntax because of easiness to parse and to map onto Frag meta models. The meta models for SOA integration (Chapter 4) focus the explanation of the UML2 Activity Diagram Meta-Model and differentiates flow models for long-running business processes and short running technical processes. The DSLS for flow models (chapter 5) is a pattern language for process oriented integration of services and can be distinguished in macro flows (long running processes) and micro flows (short-running technical processes) An example of configuration for a process-based integration architecture is given and process flow refinement, steps, macro-flow-model, micro-flow-model and macro-micro flow refinement are explained. Furthermore, the next chapter 6 (DSL for Architectural SOA Models) focuses on architectural components in the system of business object models. Again UML is used in this context: Class diagrams are used for business objects, Component diagrams are used to represent architectural abstractions. To capture semantics of a call-back architecture, the authors propose 5 stereotypes: IEvent, ICallback, EventPort, CAllbackPort, and Callback. The chapter 7 “SOA model Integration” explains the formal integration. Correlation identifiers are used to match events and call-back’s between the components. In the component model, it need to be modelled which correlation identifier as multiple identifiers can be used. In addition, it is important to ensure that macro and micro-flow-models pass a valid correlation identifier type to all asynchronous invocations. The next chapter tells about planned extensions of the model with organizational models or human-interaction models. The key criteria is the approach based on a meta-meta-model, primitives as modelling constructs, and model validation tools for these concepts. Finally, the authors conclude their paper by a quick summary of their concept.

**Comments**

The approach is very academic and not at all easy to understand for others than experts in this domain. Very specific terms are not explained and the authors presume that the reader knows about complex and technical concepts already. The focus is made on meta-meta-models, UML2, a method consisting of 7 steps for the model-driven design process, micro-macro-flow concept, and the own developed pattern language and syntax "frag". The question here is to find out how complex this is and if this can be applied without huge effort in practice.

There is so far no trace of a proven implementation or a successful case study, where this method has been applied in practice. Even, if the authors state that their approach is based on proven practices, it would be interesting to test this method in a practical environment.
However, MDA as classification criteria is not mentioned at all and it is not clear how services are defined and integrated. The micro-macro flow shows the drilling down functionality, but it is not clear how complex processes, events, actors and data are considered. Furthermore, the strategic aspect is completely neglected. The platform independent layer with business models is not discussed. It would also be interesting to see, if the mentioned tools (ARIS, ADONIS) can be used to follow the approach.

This method can certainly bring its value related to the technical aspects of model translation, verification and integration into business applications.

## Platform-independent model for service-oriented architecture (PIM4SOA)

**Principal Authors:** Xabier Larrucea et al, ATHENA Project, [BL06]
**Company/Organization:** European Software Institute (ESI) Spain, DFKI GmbH Germany, SINTEF ICT, Norway
**Year of Release:** 2006
**Category:** Whitepaper
**Web:** http://www.dsic.upv.es/workshops/dsdm06/files/dsdm06-06-Larrucea.pdf
**Nb. of pages:** 10
**Viewpoint:** functional (CIM-PIM-PSM)

**Summary:**

PIM4SOA is an open-source modelling tool with an underlying Meta model to support the design of SOA in a platform-independent (PIM) or technology neutral manner following the OMG MDA approach. The met model defines an abstract language to specify executable business processes for exchange between modelling tools and execution environments and is based on UML and EMOF. Four dimensions are covered: Service, Process, Information and Quality of service

The tool can be used within the Eclipse platform. Model transformations are available for

- PIM4SOA (UML) to PIM4SOA (EMF)
- PIM4SOA to XSD
- PIM4SOA to WSDL

- PIM4SOA to JACK

**Comments**

The development of this method seems to be in an early stage. A strength of the method might be the development based on open OMG standards UML and MDA. The method tackles in an example in a BtoB scenario the following issues:

- business processes are not defined using the same language. This barrier makes difficult the definition of a coherent and consistent process where the stakeholders have a common and unified view of the process.
- systems are not interoperable. They use proprietary format for their applications and their connections are made ad-hoc.
- functional extensibility of their applications is limited
- business processes and their systems supporting their business processes are not related in a systematic way.

The proposed method focus mainly on interoperability issues between two companies.

It will be interesting so test the method in practice and to see how this method might be re-used for the development of a practical and condensed method in chapter 2.
For the analysis, the paper A model driven approach to agent-based Service-Oriented Architecture, (Zinnikus A., Benguria G., Elvesaeter B., Fischer K., Vayssière J.)

## Service-oriented Design and Development Method (SoDD)

**Principal Authors**: Mike P- Papazoglou & Willem-Jan van der Heuvel, [PvdH06]
**Company/Organization:** INFOLAB, Department of Information Systems and Management, Tilburg University, Netherlands
**Category:** Whitepaper, Int. J. of Web Engineering and technology (IJWET), 2006
**Web:** http://infolab.uvt.nl/pub/papazogloump-2006-88.pdf
**Book: http://www.pearsoned.co.uk/bookshop/detail.asp?item=100000000029294**
**Nb. of pages:** 16
**Source:** Academic Organisation (University)
**Viewpoint:** functional & technical
**Approach**: Top-down

**Chapters:**

1. Introduction
2. Characteristics of service development Life cycle Methodology

3. Web Services Development Life Cycle Method Baseline
4. Service Oriented Design and Development Principles
5. Phases of the service oriented design and development methodology
6. the service design phase
7. Service construction phase
8. The service Provisioning Phase
9. Service development phase
10. Outlook

**Summary:**

The introduction states directly the objective of the paper: to provide an overview of the methods and techniques used in service oriented design and development and to examine a service development method from the point of view of both service producers and requesters and review the range of elements in this method that are available to them.

The second chapter explains the web service development life cycle hierarchy based on the work of IBM researchers Arsanjani and Brown. The starting point is clearly the business goals and requirements through software design, code assets and composite applications.

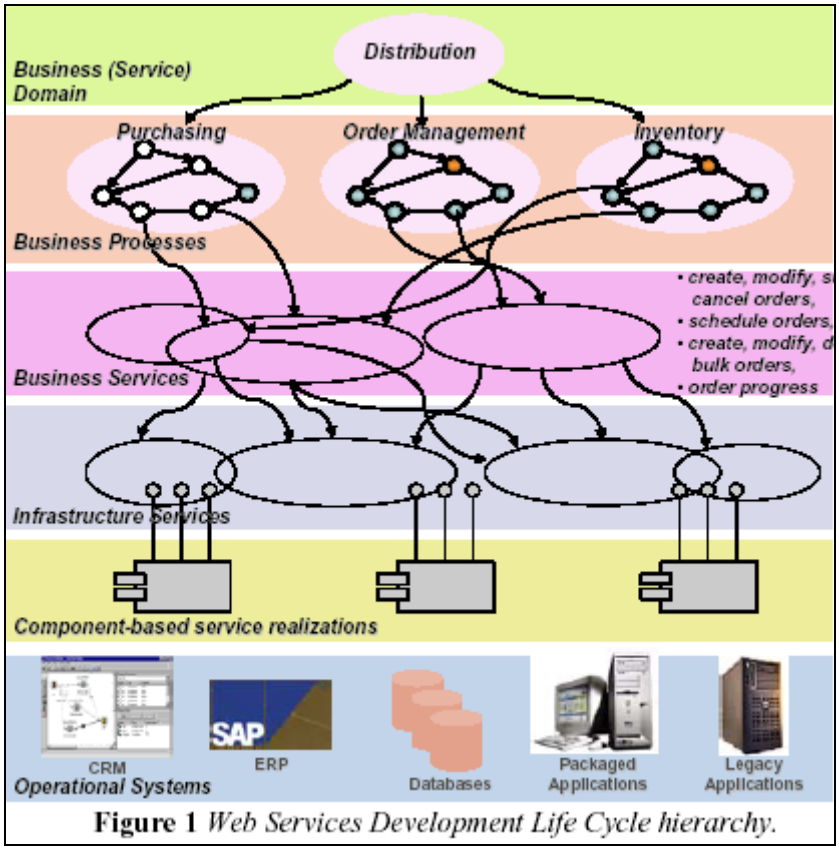



**Figure 87: Web Service Development Life Cycle Hierarchy**

The authors clearly describe the top-down approach with Business domain, Business processes, Business Services, Infrastructure Services; Component based service realizations down to operational systems.

The third chapter describes the method baseline partly based on successful models namely Rational Unifier Process (RUP, 2001, Kruchten 2004) Component based development (Herzum 2003) and Business Process Modelling (Harmon 2003). The method is an iterative and incremental process decomposed into 8 phases:

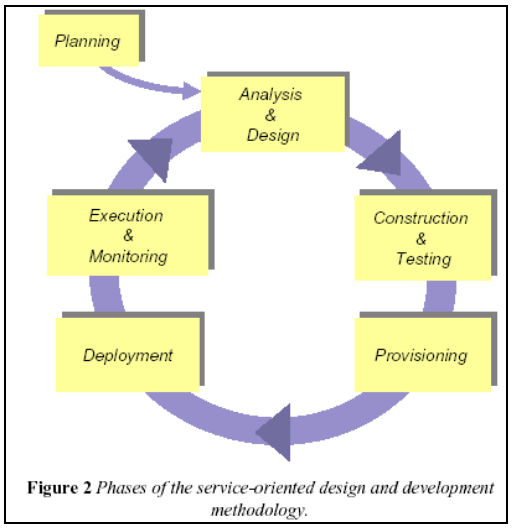



**Figure 2** *Phases of the service-oriented design and development methodology.*

**Figure 88: Phases of service-oriented design and development methodology**

During the planning phase, the project feasibility, goals, rules and procedures are set and requirements are gathered. The business case is conducted during the design phase considering various alternatives for implementing business processes, identifying web services. The next phase is service construction and testing including functional correctness, completeness and interoperability. The provisioning phase encompass issues like service metering, service rating and service billing. The service deployment and advertisement is done through the repository system. The final phase includes execution and monitoring of web services.

The fourth chapter focus on key principles which are 1) service coupling and 2) Service cohesion 3) Service granularity. The three principles are explained and recommendations are given.

Chapter 5 focus again on the first of the 8 phases described in short before. This time is explained more deeply. The analysis phase encourages a radical view of process (re)-design and supports the re-engineering of business processes. Its main objective is the reuse of business process functionality in new composite applications. To achieve this objective the analysis phase comprises four main activities: process identification, process scoping, business gap analysis, and process realization.

Chapter 6 describes the service design phase with concerns 1) Component granularity 2.) service re-usability 3.) Service composability. Then, it is said how services should be specified including service interfaces, messaging and coupling. WSDL is used to show operation parameters and how services should be programmed. Service policy concerns including Quality of service issue are explained and for the service orchestration, BPEL is recommended. The author also recommends tools such as IBMs WebSphere Business Modeler to perform analysis, what if simulation to estimate business benefits and the transformation into UML and BPEL models. The authors are also highlighting the BPMN notation as a standard to define unambiguously business logic and information requirements.

Non-functional business process requirements such as performance, payment model, security model and, transactional behaviour. These concerns are explained through a Service Level Agreement (SLA) example.

Chapter 7 explains very quickly the service construction without going into details.

Chapter 8 describes briefly two methods of testing 1) dynamic testing and 2) functional testing. (Brown 2002). The testing includes performance, response times, transaction rates, stress testing, interface testing, assembly testing, network congestion, security, and upgrade tests. The objective of the testing is to make sure service requirements such as privacy, message integrity; authentication, authorization and non-repudiation are met.

Chapter 9 "The service provisioning phase" is central to operating revenue generating web services between organisations. Aspects such as service governance, service certification, service enrolment, service auditing, metering, billing and mapping needs to worked on.

The last 3 chapters are again very brief and provide an overview what is meant by the service deployment, service execution and service monitoring

The outlook finally sums up the introduced method and states that the authors will gather real-life case studies in different sectors and develop an own toolset to effectively support the methodology.

**Comments:**

The whitepaper positions itself with the objective to "…provide an overview of the methods and techniques used in service-oriented design and development". Indeed, one method has been chosen (IBMs SOMA, Arsanjani & Brown) and has been enhanced by the authors. Unfortunately, there is no comparison to other methods. However, the paper explains well for functional profiles the method and starts by defining the SOA infrastructure hierarchy. The structure of the phases is the classical approach (RUP, Component based development, Business Process Modelling), which make sense to apply it also for web-service developments. Key principles as the foundation for service based process design are well explained. The planning and design phase gives important information about scoping of processes, how processes can be identified and the different realization options (Brittenham 2001) 1.) Green field development, 2.) Top-down development 3.) Bottom-up development, and 4.) Meet-in-the-middle developments are explained. I do not agree with the issues stated for the options 2 to 4 being ambiguous related the priorisation of the processes to start with. It depends rather on the context of the specific organization to apply the method with the best fit. The authors are giving reference models as solution e.g. RosettaNets standard processes. Normally, priorities can well be set up relating to the conducted business case in the planning phase.

Furthermore, service design concerns are well explained and how services could be specified. The authors do not mention the MDA method nor is their focus to give an overview of models that could be used other than to use WSDL for services and BPEL for orchestration. Furthermore, they name BPMN as a business process language. UML or any other business process modelling language does not play any role whilst the authors focus on the importance of process modelling, design, analysis etc. The strategy phase, strategy concepts and methods are also not taken into consideration

Overall it is a well-structured method based on SOMA (IBM) explaining well the critical concepts and success factors for service development. The authors have so far not gathered practical experience with their enhanced SOMA method, but this could be an interesting area to see in future. The intent to develop an integrated toolset to effectively support the method needs to be monitored carefully. It is not said, if these tools should be supported by software.

## Service oriented Modeling & Architecture (SOMA)

**Principal Authors**: Ali Arsanjani, [Ars04]
**Company/Organization:** IBM
**Category:** Article, developerworks, 2004
**Web:** http://www.ibm.com/developerworks/library/ws-soa-design1/
**Nb of pages:** 10
**Source:** Commercial Organisation
**Viewpoint:** functional
**Approach**: Top-down

**Chapters:**

1.) Introduction
2.) SOA a conceptual model
3.) The architectural style and principles
4.) Context
5.) An architectural template for SOA
6.) How to approach service-oriented modelling and architecture
7.) Service-oriented modelling: The analysis and design of services
8.) Conclusion

**Summary:**

The objective of the article describing the SOMA method contains techniques required for the identification, specification and realization of services, their flows and composition, as well as the enterprise-scale components needed to realize and ensure the quality of services required of an SOA.

In the introduction, the author states that SOA is not a product but more about business-aligned IT services using a set of design principles, patterns, and techniques. SOMA is enhancing the object-oriented analysis and design (OOAD) by addressing services, flows, and components.

The conceptual model in chapter 2 describes very brief the link between Service consumer, service provider and service broker. Chapter 3 focuses briefly on the SOA benefit such as business agility and defines what a web service is.

In chapter 4, Arsanjani says that the context in which the company is plays a key role. Therefore a maturity model can help. When starting a SOA project, assessments with eventually pilots should be done. Important is also strategy and planning activities including

migrating plans, tools, methods, training, technologies, standards, roadmap, governance and implementation of best practices (security, performance, compliance with standards for interoperability, change management).

Chapter 5 explains the layers of SOA: presentation, Business Process Choreography, Service, Enterprise Components, Operational Systems, Integration Architecture, and QoS, Security, Management and Monitoring

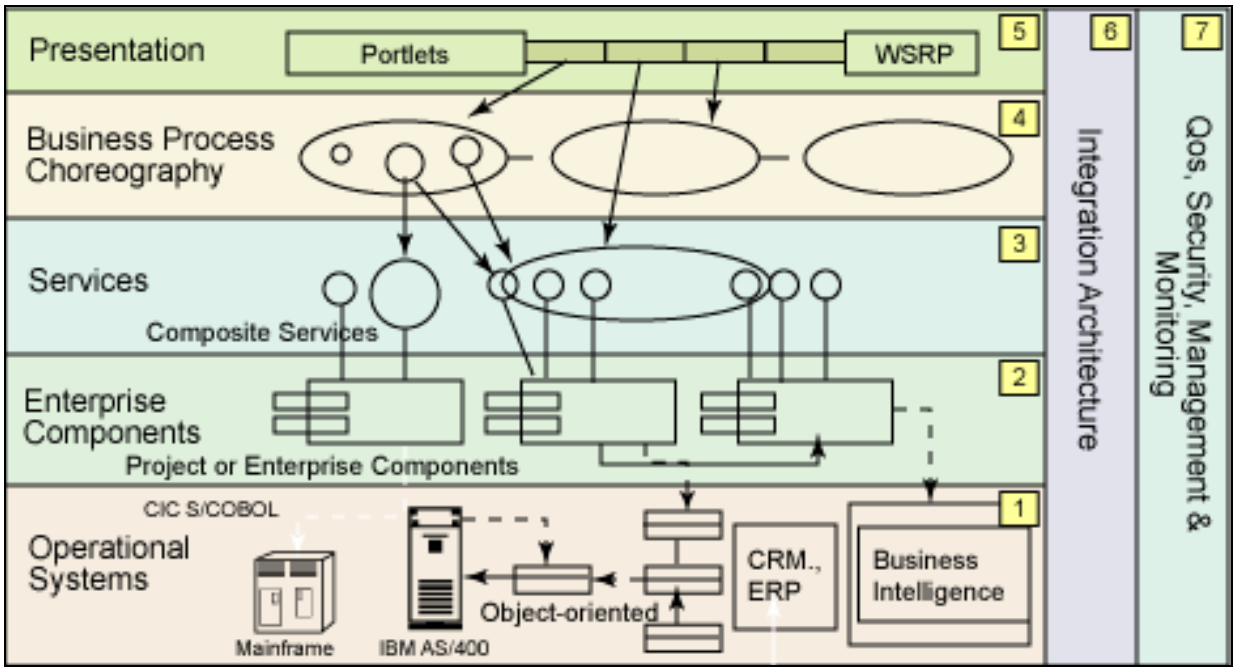



**Figure 89: The layers of a SOA in SOMA Methodology**

The different layers are explained in brief.

The next chapter" how to approach service oriented modelling and architecture" describes how to combine a top-down, business driven approach with a bottom-up approach. The best approach seems to be first top-down, then goal-service modelling, and finally bottom-up legacy analysis of existing assets. The faster the project is scoped down to a manageable realistic set; the sooner value by focusing on key services can be achieved.

The next chapter dedicated to design and analysis says that SOA is more strategic and business aligned. Web services are a tactical implementation of SOA. Arsanjani talks about roles and activities of Service providers and service consumers. The service identification in the top-down view includes a blueprint of business use cases for business services e.g. a domain composition, which consists of the decomposition of the business domain into its functional areas and subsystems, including its flow or process decomposition into processes, sub-processes, and high level business use cases.

The bottom-up method is used for existing system analysis. Service candidates are identified in order to analyse and leverage API's, transactions, and modules from the legacy and packed applications.

The middle-out view consists of goal-service modelling to validate other services not captured either by top-down or bottom-up service identification approaches. It ties services to goals and sub-goals, key performance indicators, and metrics.

Then services are classified and categorized, then a subsystem analysis is performed. Components are specified such as Data, Rules, Services, Messaging, event specification configurable profile, and variations.

Services are then allocated to identified subsystems and realized: Services are integrated and transformed. Here the following mix of approached is recommended: Top-down domain decomposition (process modelling and decomposition, business rules analysis, and domain specific behaviour modelling). Bottom-up should be done in parallel analysing existing legacy assets that are candidates for componentization and service exposure. To catch the business intent behind the project and to align services with the business intent, goal service modelling is conducted.

The conclusion sums up but underline the importance of the combination of the three approaches top-down , bottom-up, and goal model analysis (middle-out) should be done.

**Comments:**

The approach is well structured and based on IBM´s best practice from projects. The most interesting part of this method is the combination of different approaches (top-down, bottom-up, middle-out). Proven methods like object oriented analysis and design (OOAD) are re-used and adapted to the SOA requirements. However, some chapters are really short, it is said what to do but not how. There is no link to MDA or types of models and tools that could be used. The method seems to be well known in the practice and academia world, as IBM was one of the first commercial organizations to create an own method. The success of the method has also influenced Papazoglou for the enhancement of his proposed method SODD (chapter 5.8.) It will be interesting to see in the empiric research how successful this method is in practice.

## ORACLE Unified Method for SOA (Version 5.5.)

**Principal Authors**: Adam Korczak, Girish Krishnan, Piotr Skrobisz, Stephen Verba, Stephen Bennett, Sigrid Gylseth, Jan Kettenis [ORAC11] , based on former method BEA[Shu06]

**Company/Organization:** ORACLE
**Category:** Framework Tool, 2011
**Web:** OUM is restricted access and not available on the web.
**Nb. of pages or Size:** 92MB Browser Tool Framework, 2396 Pages
**Source:** Commercial Organisation
**Viewpoint:** functional
**Approach**: Top-down

**Chapters or Structure:**

1. SOA Program Scope Engagement (typically, at the enterprise-level) - The tasks for these type engagements are found in the OUM Envision focus area.
2. SOA Project Scope Engagements - The tasks for these type engagements are found in the OUM Implement focus area.
3. The Service-Oriented Architecture (SOA) Core Workflow view - is used to provide a conceptual view of the SOA approach that is provided by OUM.

**Summary:**

Service-Oriented Architecture (SOA) in OUM covers the entire lifecycle for services. It is important to have an overall picture of the different dimensions for planning and delivering SOA. SOA efforts may vary in their scope and level of effort and the approach taken. OUM supports all these dimensions across the Envision and Implement focus areas.

1. Roadmap Creation, which focuses on assessing the current state of the enterprise in respect to their SOA goals and the maturity of the capabilities, required to execute SOA successfully. The tasks to support Roadmap Creation can be found in the Envision focus area.

2. Strategy and Planning, which concentrates on defining a number of key frameworks. The tasks to support Strategy and Planning can be found in the following views:

    - SOA Engineering Planning
    - SOA Modeling Planning
    - SOA Governance Planning
    - SOA Reference Architecture Planning

The main artefacts at the end of a program scope engagement are an incremental SOA implementation roadmap that maps out the build-out of the infrastructure, the solution roadmap and services roadmap.

For engagements with a project scope, enterprises start to execute their incremental SOA implementation roadmap and start to deliver value to the business. Such project engagements cover the different lifecycles of delivery of solutions and delivery of services and the associated service infrastructure. The tasks to support project scope can be found in the Implement focus area.

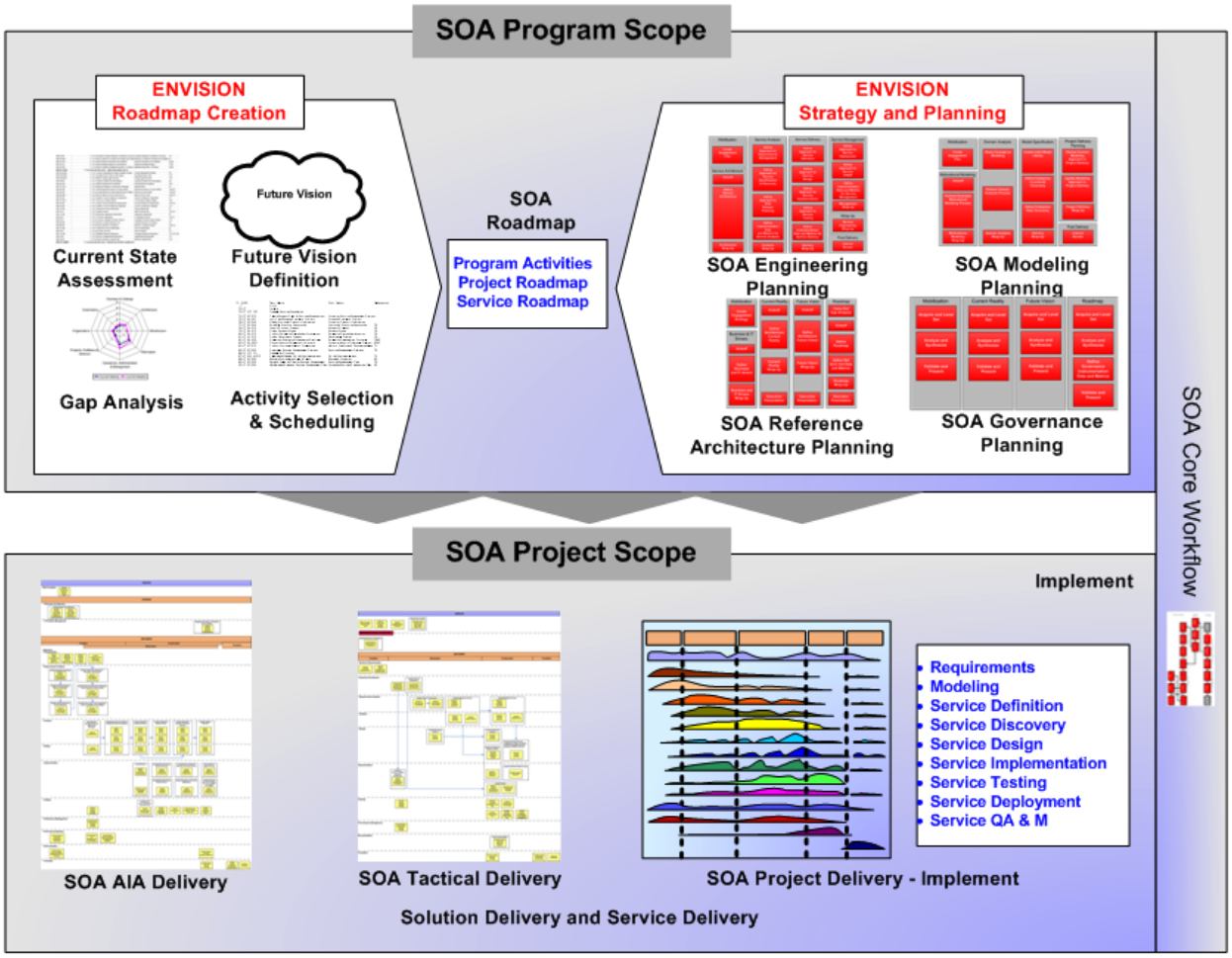



**Figure 90: OUM for SOA Overview**

3. The ***SOA Core Workflow*** describes the sequential method of OUM for SOA:

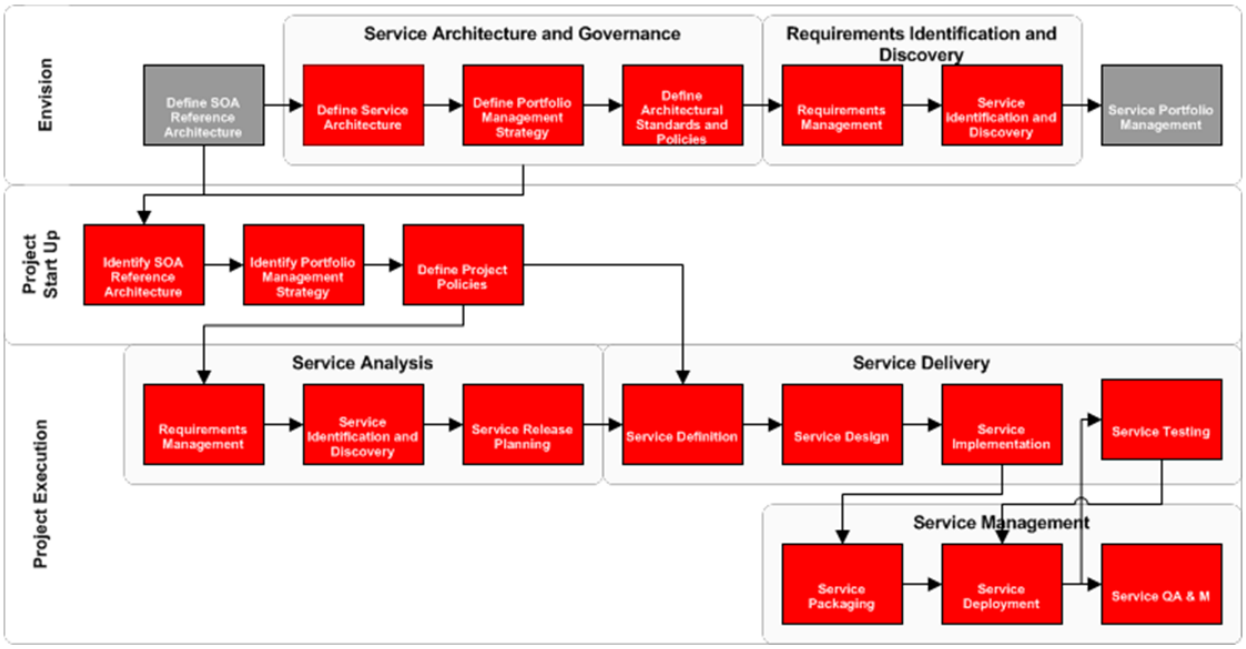



**Figure 91: OUM SOA Core Workflow**

This view is intended to describe how OUM supports SOA from Envision through Implement. This view is not used to deliver an engagement, but rather to describe at a high-level, the work that is done to prepare an enterprise for SOA. The view highlights the service engineering tasks of OUM and explains how they relate to each other.

Every box is detailed with different activities to perform and expected work products. Templates and examples are provided for some activities. As the framework is very comprehensive, we will focus on the modeling part of the method. Therefore, the activity of "functional or process modelling" is explained.

## FUNCTIONAL OR PROCESS MODELING OVERVIEW

**Table 49: Functional or Process Modeling Overview**

| No. | Step | Component | Description |
|---|---|---|---|
| 1 | Determine functional or business process levels. | Functional/Business Process Levels | There are multiple possible representations of the different levels a process or Function Model can have. The functional/business process levels describe the number of levels that you will use in the enterprise, and what each level represents. |
| 2 | Determine modeling techniques. | Modeling Techniques | The Modeling Techniques describe what kind of techniques should be used to model the different functional/business process levels. Most often, different techniques will be used to model the higher and the lower levels. |
| 3 | Determine enterprise modeling notations. | Enterprise Modeling Notations | The Enterprise Modeling Notations describe the modeling notations that should be used for the different Modeling Techniques used in the enterprise. |
| 4 | Determine the upper level Function or Process Models. | Upper Level Functional or Process Models | The Upper Level Function or Process Models are the actual models for the enterprise, in level 0 and 1 (according to pyramid B). |
| 5 | Link enterprise requirements to functions/processes. | N/A | If requirements are maintained at enterprise level, you should use the Enterprise Function Model or Process Model and tie the requirements to the appropriate location in the model. |

The following levels are defined:

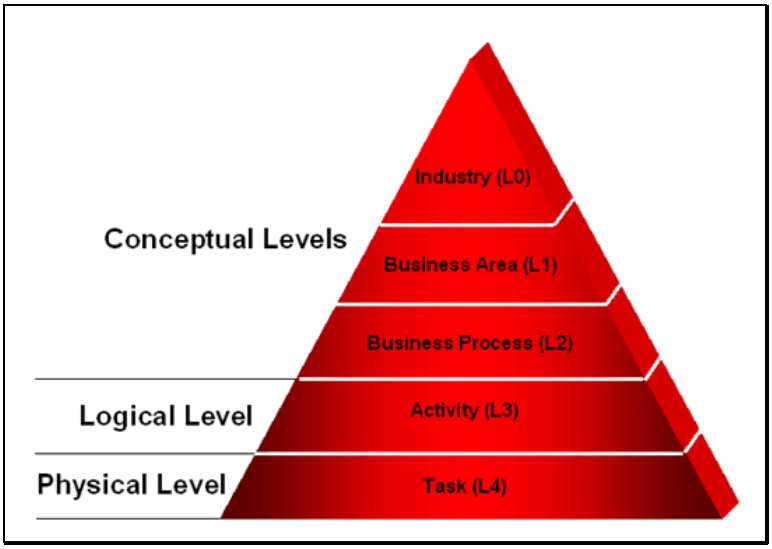



**Figure 92: ORACLE Levelling for Process-Notations**

An example of Process levelling in OUM SOA:

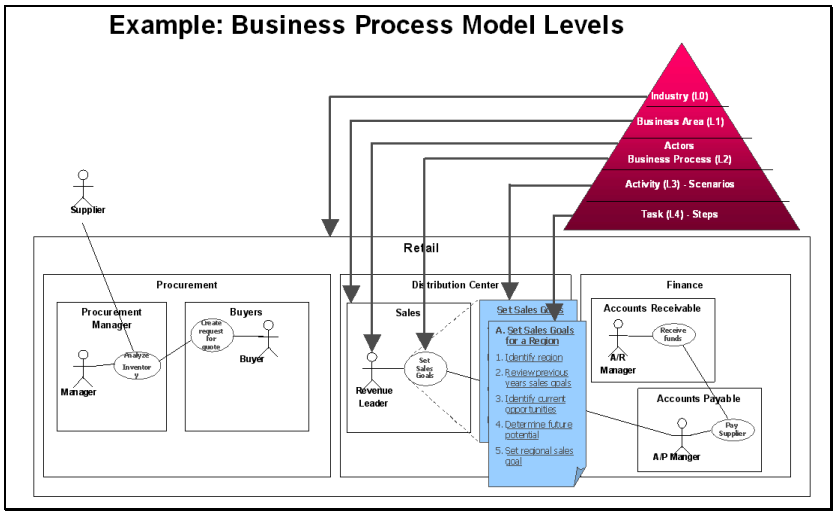



**Figure 93: Example of ORACLE Levelling for Process Notations Using UML**

To determine an improved future process, begin by working with business analysts and end users analysing an "as-Is" process and capturing the issues and challenges facing this process, i.e., delays, disconnects, etc. A business process can be decomposed into several sub-processes, which have their own attributes, but also contribute to achieving the goal of the super-process. The analysis of business processes typically includes the mapping of processes and sub-processes down to activity level.

It is often easier to first model the as-is process (if not already available), before thinking about the improvements.

You could also start with doing a functional analysis of the enterprise in levels similar to the way it is described here. For each level, you could use a different modeling approach.

For example a **Value-Added Chain Diagram (VACD)** is typically used for levels up to business process (level 2). Other modeling approaches, such as **BPMN (Business Process**

**Modeling Notation)** are better to use for modeling the level 3 and below. Alternatively, you could use **UML activity diagrams**.

VACD (Value-Added Chain Diagrams) is a less formal approach as it has a rather informal notation standard that allows for deviations from "text-book" notation and inclusion of non-standardized symbols. BPMN (Business Process Modeling Notation) is standardized, as is UML activity diagramming. Because of its informal nature, VACD might be easier for business people to understand but tends to be less precise and as a result harder to map on analysis and design models

To model the lower level business processes, you may choose to use BPMN, UML activity diagrams, or some other notation.

BPMN is formalized to a level where there is a clear mapping to BPEL (Business Execution Process Language). For example, a tool such as the Oracle BPA Suite can do a mapping from BPMN to BPEL automatically (to some extent). However, the client may already have standardized on a specific modeling approach. If so, consider using that approach in that it is well known to the client.

In case of BPMN or UML activity diagrams, the diagram shows the events (initial state), the steps and decision points as an actor performs them, and their sequence, by drawing flows between pairs of activities or between an activity and a decision point or join. When there are decision points that split a process into more directions, first identify the main flow before going into all the exceptions routes.

An example of BPMN business process model with horizontal swimlanes is shown below:

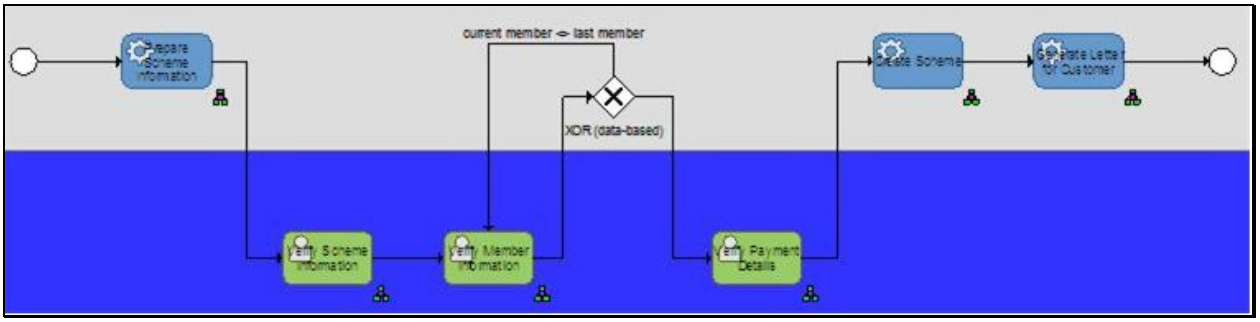



**Figure 94: ORACLE SOA BPMN Example**

When getting down to the requirements analysis (refer to the Enterprise Requirements Management technique), groups of requirements can be attached at level 2 (Business Process) and below. Requirements encountered at a parent node need to be interpreted that they apply at the child levels (and grand child, and so on) and below. This is an especially useful method for scoping or establishing a hierarchy of requirements which can be disseminated into service candidates as well as ensuring complete coverage. It is also a useful way to broadcast non-functional requirements (rather than repeating them at every node in every child level). Further, this strategy can be used to document higher granular requirements prior to breaking them down in to finer grained requirements.

## FUNCTIONAL MODELING AND SOA

A Function Model that prevents duplication of enterprise functionality across the model is one of the aspects that make it useful for application within SOA deployments. One of the major issues that the SOA strategy attempts to overcome is the duplication of critical business function across systems. In many cases this happens simply because there is a lack of visibility with respect to requirements and existing IT business function. In other cases the inter-departmental rivalries/differences are the cause. A Function Model that eliminates functional duplication and is scalable with respect to enterprise class data sizes is an ideal fit for supporting the service identification and discovery aspects of an SOA from a functional point of view.

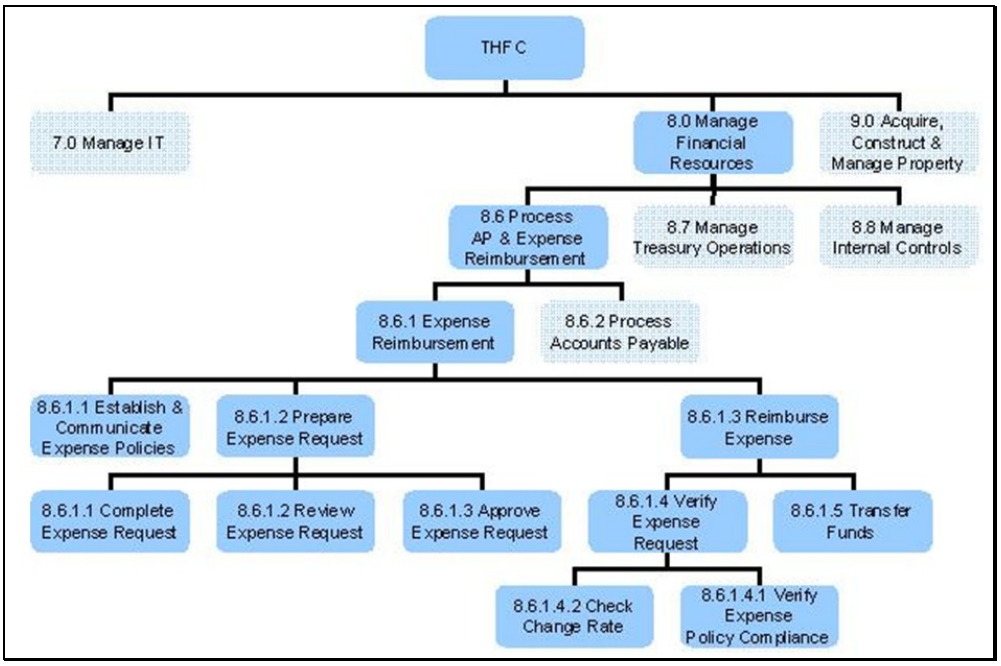



**Figure 95: ORACLE SOA Functional Modeling Example**

## RULES THAT APPLY TO FUNCTION MODELS

- Function models are trees and as such, the following rules apply to them:
- Function Models have a fixed number of functional levels. However the detail incorporated within each level can vary.
- Each decreasing level is finer grained with respect to functional representation when compared to the level(s) above.
- Each node in the graph may have exactly one parent (excluding the root node, which has no parent).
- Cycles are not allowed in the model.
- A Function Model is navigated by narrowing functional granularity, and not by organization structure.

## QUALITY CRITERIA

Use the following criteria to check the quality of this technique:

- Is there a clear definition on what kind of functional or process levels should be used to model the enterprise?
- Is there a clear statement on what kind of modeling techniques should be used for the different levels or situations?
- Is it clear what kind of notation should be used for each modeling technique?
- Is a high level model created in the modeling technique prescribed for the highest level, and according to the prescribed modeling notation?

Templates or other Examples are not available.

**Comments:**

This framework is well structured, derived from the Unified Process (UP), and much "best practice" oriented. The audience is functional oriented and written in clear and easy-to-understand text. The description is not adapted to a specific industry or customer type. The decomposition of the framework into program scope and project scope separates the strategic reflection from the SOA implementation project. The "Envision" part explains in detail how to create a SOA Roadmap based on Maturity Assessment. Next, the "SOA Planning" dimensions (Engineering, Modelling, Architecture, Governance) give method guidance on how to perform the preparatory work for the SOA implementation project. Once the SOA program is established, the SOA project implementation method details the SOA application integration Architecture, the SOA tactical and the SOA project delivery.

The SOA Core Workflow presents a sequential order of macro processes with related activities to explain details. This part of the framework is within the scope of SOA implementation method as defined in this thesis.

The framework offers for each part activity description and in some cases also with concrete examples. All the recommended tools are only ORACLE family tools and the examples and templates are not necessarily added value. The details on notation usage can be summarized to VACD, UML Scenario, BPMN and BPEL. The transition between the levels is inherent to the proposed notations, which are supported by ORACLE products.

In total, the latest version is very comprehensive and complete method framework, where SOA is one of five different scenarios (BI/Enterprise Performance Management, BPM, CRM, Software Implementation and SOA). However, ORACLE has weaknesses in harmonizing the wide range of different tools and related methods acquired during the last years. This becomes clear in the proposed framework, as links between topics, phases and tool integration seem not to be smooth and mature.

## SOA Adoption Model

**Principal Authors**: -
**Company/Organization:** GARTNER, [GART06]
**Category:** Guideline, SOA Adoption Model, Gartner Leader's Toolkit 2006
**Web: -**
**Nb. of pages:** 8
**Source:** Commercial Research Organisation
**Viewpoint:** functional
**Approach**: -

**Chapters:**

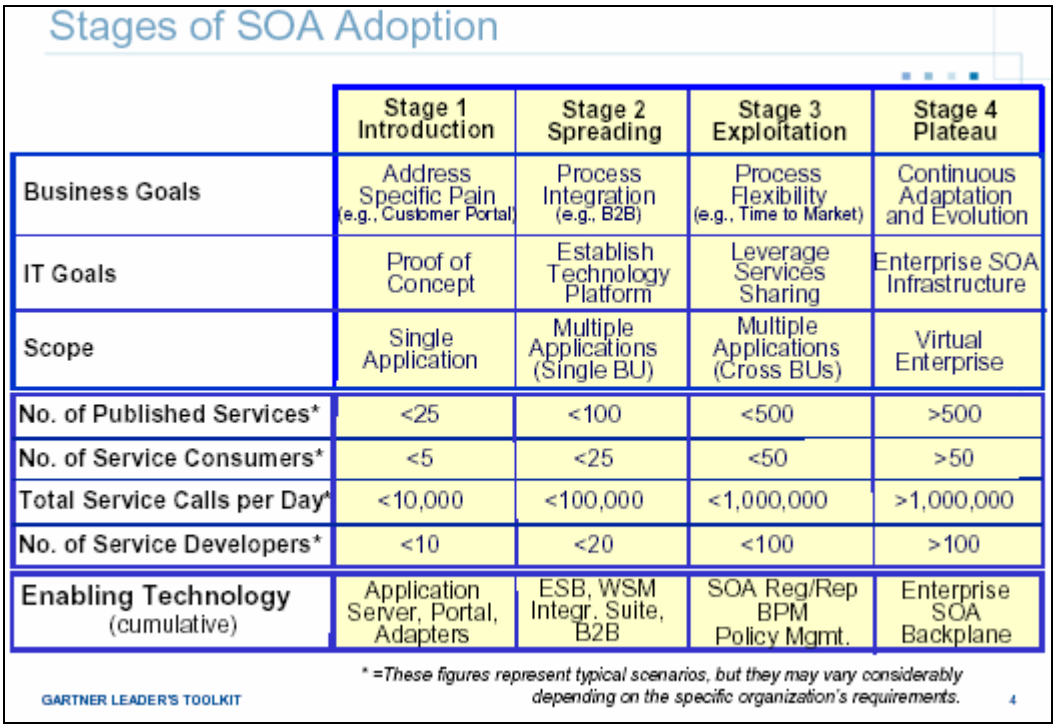
1.) Adopting SOA: Business & IT Drivers
2.) Benefits and Implications
3.) Stages of SOA Adoption
4.) Required Management Buy-In per Stage
5.) Required Technology Skills per Stage
6.) Required Capabilities per Stage
7.) Recommended Approach

**Content:**

The first chapter differentiates between "Top-down" Enterprise drivers (M&A, Multichannel sales support, Time to Market etc…) "Bottom-up" Business Unit drivers (Call centre integration, process integration, real-time B2B etc.) and "Perennial" IT challenges ("doing more with less", Business/IT alignment, Data consistency/quality etc.)

The second chapter lists benefits (architectural partitioning, incremental deployment, sharing/Reuse of Services) and implications (Higher Upfront cost, more distributed infrastructure, tighter management/governance)

The third chapter describes the 4 stages of SOA adoption:



Stages of SOA Adoption

| | Stage 1 Introduction | Stage 2 Spreading | Stage 3 Exploitation | Stage 4 Plateau |
|---|---|---|---|---|
| Business Goals | Address Specific Pain (e.g., Customer Portal) | Process Integration (e.g., B2B) | Process Flexibility (e.g., Time to Market) | Continuous Adaptation and Evolution |
| IT Goals | Proof of Concept | Establish Technology Platform | Leverage Services Sharing | Enterprise SOA Infrastructure |
| Scope | Single Application | Multiple Applications (Single BU) | Multiple Applications (Cross BUs) | Virtual Enterprise |
| No. of Published Services* | <25 | <100 | <500 | >500 |
| No. of Service Consumers* | <5 | <25 | <50 | >50 |
| Total Service Calls per Day* | <10,000 | <100,000 | <1,000,000 | >1,000,000 |
| No. of Service Developers* | <10 | <20 | <100 | >100 |
| Enabling Technology (cumulative) | Application Server, Portal, Adapters | ESB, WSM Integr. Suite, B2B | SOA Reg/Rep BPM Policy Mgmt. | Enterprise SOA Backplane |

** =These figures represent typical scenarios, but they may vary considerably depending on the specific organization's requirements.*

GARTNER LEADER'S TOOLKIT 4

**Figure 96: Gartner SOA Adoption Model Overview**

Chapter 4 describes the roles and their implications into each stage:

| | Stage 1 Introduction | Stage 2 Spreading | Stage 3 Exploitation | Stage 4 Plateau |
|---|---|---|---|---|
| Head of Development or Head of Integration | ✓ | ✓ | ✓ | ✓ |
| CTO/Head of Architecture | × | ✓ | ✓ | ✓ |
| Head of IT Operations | | × | ✓ | ✓ |
| CIO/Business Units | | × | ✓ | ✓ |
| CEO | | | × | ✓ |

**Figure 97: Gartner SOA Adoption Model Implications**

Chapter 5 defines the required technology skills per stage:

| | Stage 1 Introduction | Stage 2 Spreading | Stage 3 Exploitation | Stage 4 Plateau |
|---|---|---|---|---|
| Basic Middleware | ✓ | ✓ | ✓ | ✓ |
| Web Services | ✓ | ✓ | ✓ | ✓ |
| Integration Middleware | × | ✓ | ✓ | ✓ |
| Service-Oriented Development of Applications (SODA) | | × | ✓ | ✓ |
| Business Process Management | | × | ✓ | ✓ |
| SOA Operations Management | | × | ✓ | ✓ |

✓ = Imperative
× = Recommended

**Figure 98: Gartner SOA Adoption Model Implications Details**

Chapter 6 defines the required technology skills per stage:

| | Stage 1 Introduction | Stage 2 Spreading | Stage 3 Exploitation | Stage 4 Plateau |
|---|---|---|---|---|
| SOA Center of Execellence | × | ✓ | ✓ | ✓ |
| Service Life Cycle Management | | × | ✓ | ✓ |
| Service Design Methodology | | × | ✓ | ✓ |
| Planning Control and Quality Management | | × | ✓ | ✓ |
| Service Reuse Methodology | | × | ✓ | ✓ |
| Operation Management | | × | ✓ | ✓ |
| Domains | | | × | ✓ |
| Cost Allocation Schema | | | × | ✓ |
| Consistent Enterprisewide Governance Processes | | | × | ✓ |
| Enterprisewide SOA Backplane | | | × | ✓ |

**Figure 99: Gartner SOA Adoption Model Implications Skills**

Finally, chapter 7 sums up and gives 8 steps as recommended approach:

1. Define target SOA adoption stage
2. Assess the in-place SOA-enabling technology portfolio
3. Assess the available technology skill set
4. Assess available capabilities
5. Perform a gap analysis
6. Define a plan to fill gaps
7. Get required management buy-in
8. Implement the plan

**Comments:**

The Gartner adoption model is a mixture between maturity model and approach. On the one hand, stages are defined describing an incremental approach. The categorization of different drivers in "top-down" , "bottom-up" and "IT challenges" is an interesting view on strategic objectives. Following the Balanced Scorecard method, the objectives are structured into 4 views with bottom-up relationships. Gartner also recommends a top-down approach starting with business objectives and a careful Return on investment calculation which is not as easy as Gartner is saying. The step-by-step approach takes into consideration a change mgt. as organizational and technical challenges need to be addressed. Without mentioning "Governance", Gartner is explaining parts of it by defining required roles& responsibilities, technology, skills, and capabilities per stage. As the stages with their scenarios are not described in detail, it is difficult to evaluate, if the recommendations are right. This is hard to verify, as the technology and skills are not explained e.g. Service-oriented development of applications, SOA operation management, Service life cycle management etc. There is no link to MDA and model types to use. BPM is a recommended skill to be considered as from stage 2.

Overall the recommended approach is a high-level attempt using the classic top-down incremental approach to structure the main activities through the introduction of SOA. This approach could be complementary to more technical and comprehensive approaches.

## SOA Delivering Strategies

**Principal Authors**: Thomas Erl: SOA – Concepts, Technology and Design, Chapman Hall 2006, [NL05]
**Company/Organization:** -
**Category:** Book
**Web: -**
**Nb. of pages:** 8
**Source:** Commercial Author
**Viewpoint:** functional
**Approach**: -

Chapters

1.) Introduction
2.) Case Studies
3.) Introducing SOA
4.) The Evolution of SOA
5.) Web Services and Primitive SOA
6.) Web Services and Contemporary SOA (Activity Management & Composition)
7.) Web Services and Contemporary SOA (Advanced Messaging, Metadata, Security)
8.) Principles of Service Orientation
9.) Service Layers
10.) SOA Delivery Strategies
11.) SOA Analysis: Introduction
12.) SOA Analysis: Service Modeling
13.) SOA Design: Introduction

14.) SOA Design: Composition Guidelines
15.) SOA Design: Service Design
16.) SOA Design: Business Process Design
17.) Fundamental Web Services Extensions
18.) SOA Platforms
19.) Appendix A: Case Studies conclusions
20.) Appendix B: Service Model Reference

**Content:**

Pls. refer to Book.

**Comment:**

The relevant chapters describing the method are 10 to 16. Erl describes 3 method scenarios: "top-down", "bottom-up", and "agile" strategies.

The "top-down" strategy promotes the formal definition of corporate business models prior to modelling service boundaries and can result in the highest quality level of SOA. It imposes also a significant volume of up-front analysis work.

The "bottom-up" strategy is not considered as a strategy at all, because this makes just sense when adding web services to their existing application environments. Neither Service orientation principles nor business strategy can be considered in the right way.

The "agile" strategy proposes a combination of top-down and bottom-up, where on-going analysis is supported, while allowing the immediate delivery of services. As analysis progresses, existing services are revisited and revised as required.

Erl also states clearly, that a SOA without clear business objectives will fail. Erl is stating that, but is not showing how this could work and what methods (BSC, Value Chain) or model types to use. Erl is showing in his approach what should be done and how, but he is more focussing on explaining in detail the concepts of services (WSDL), orchestration (BPEL), and messaging (SOAP).

He is not mentioning MDA levels of abstractions nor related model types and interoperability. In his practical examples, he is using ERM, BPML and UML charts, but does not explain that the chart is based on BPMN or UML language.

In total, the method is helpful on technical aspects, but is by far not enough for a complete and comprehensive method.

## SOA Organizational Roadmap

**Principal Authors**: Dirk Krafzig et al: Enterprise SOA book, [PT06]
**Company/Organization:** -
**Category:** Book
**Web: -**
**Nb. of pages:**

**Source:** Commercial Author
**Type:** functional
**Approach**: -

**Chapters:**

1.) An Enterprise IT Renovation Roadmap
2.) Evolution of the service concept
3.) Inventory of distributed computing concepts
4.) Service Oriented Architecture
5.) Services as building blocks
6.) The architectural Roadmap
7.) SOA and Business Process Management
8.) Managing Process Integrity
9.) Infrastructure of the Service Bus
10.) SOA in Action
11.) Motivation & Benefits
12.) Organizational SOA Roadmap
13.) SOA –driven Project Management
14.) RealWorld experience: Deutsche Post AG, Winterthur, Credit Suisse, Halifax

**Content:**

Pls. refer to book

**Summary:**

Krafzig et al describes in chapter 7 well the importance of BPM related to the service oriented architecture. He is mentioning MDA and CASE (Computer Aided Software Design) as methods as part of Business Process Management Systems (BPMS). Modeling languages are mentioned in chapter 7.1.3.1 such as BPEL, XLANG, WSFL, BPMN, UML, PetriNet, but the authors are not putting the languages into the MDA context. UML is mentioned and MDA is proposed as preferred approach for transformation of models from one level of detail to another.

Related to the strategy of the method, the authors opt clearly for a top-down method for service design to ensure that all service definitions meet business requirements, are designed on the right level of granularity, provide potential for re-use, ensure scalability and integrity, independent from any underlying implementation and provide appropriate service level specifications.

The presented approach starts with the objectives and benefits, but not in a structured, model-driven way. Interesting is the description of the different motivation and challenges of different roles for a SOA introduction (CEO, CIO, Architect, Project Manager, Functional Department, Developer). In reality, it is not an approach, as this would require phases, which are not described here.

Some thoughts about project management, SOA Governance and do's and don'ts are useful, but rather generic. The statement to just take a generic project method and to enhance it with service orientation components is also rather too generic. I agree to use an incremental

approach, such as other authors of methods, e.g. the “thin thread model”, which is an iterative development methodology.

## Reference Model for Service Oriented Architecture 1.0

**Principal Authors**: MacKenzie M. (Adobe Systems), Laskey. K. (MITRE Corporation), McCabe F. (Fujitsu), Brown P. (Justbrown.net), Metz R. (Booz Allen Hamilton), [MacK06]
**Company/Organization:** OASIS
**Category:** Reference Model, Document Identifier: wd-soa-rm-cd1 10. February 2006
**Web:** http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm
**Nb. of pages:** 28
**Source:** Standards Organisation
**Type:** functional
**Approach**: -

**Chapters:**

1.) Introduction
2.) Service Oriented Architecture
3.) The reference model
4.) Conformance Guidelines
5.) References
6.) Glossary
7.) Acknowledgements

**Summary:**

In the introduction, the goal of the reference model is to define the essence of service oriented architecture, and emerge with a vocabulary and a common understanding of SOA. First, the reference model is related to other work:

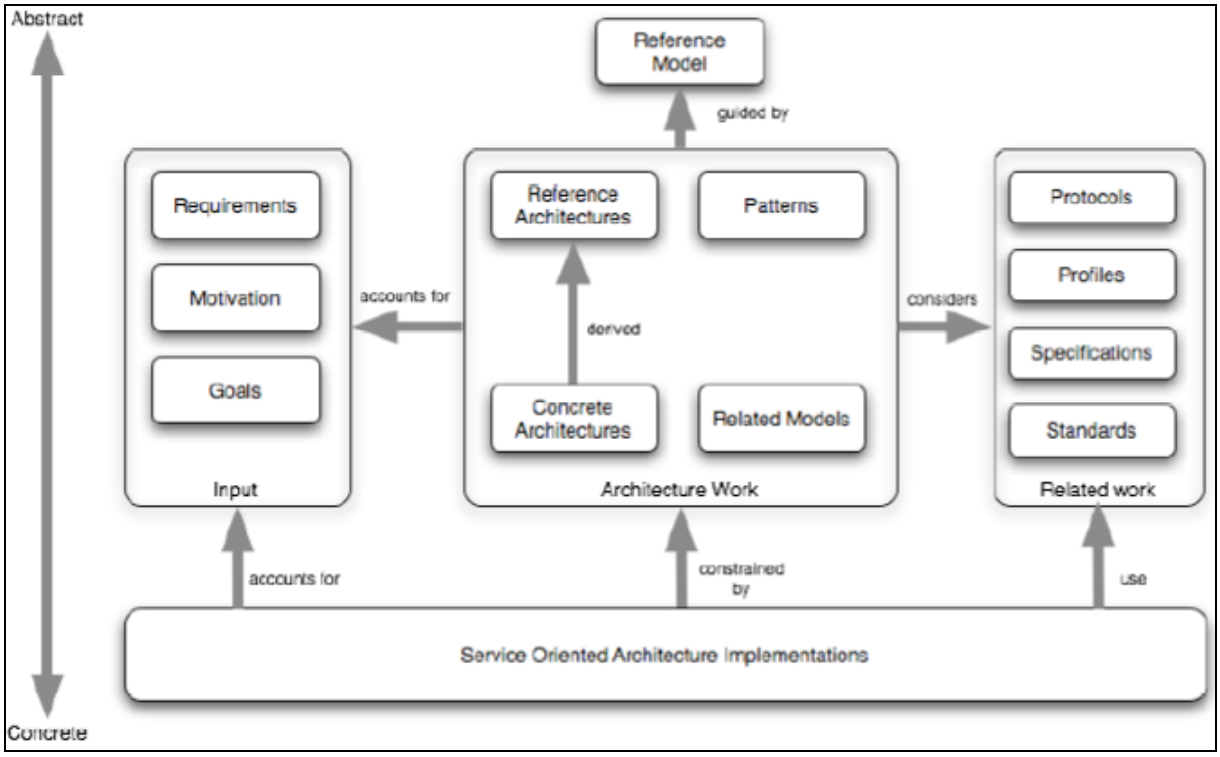



**Figure 100: OASIS SOA Reference Model**

In chapter 2, the SOA paradigm is explained. Services, Service providers, service consumers, service participants, service descriptions and service interfaces are introduced.

Chapter 3 introduces the reference model. First, the principal concepts are listed. The relationships between the concepts are developed through the paper:

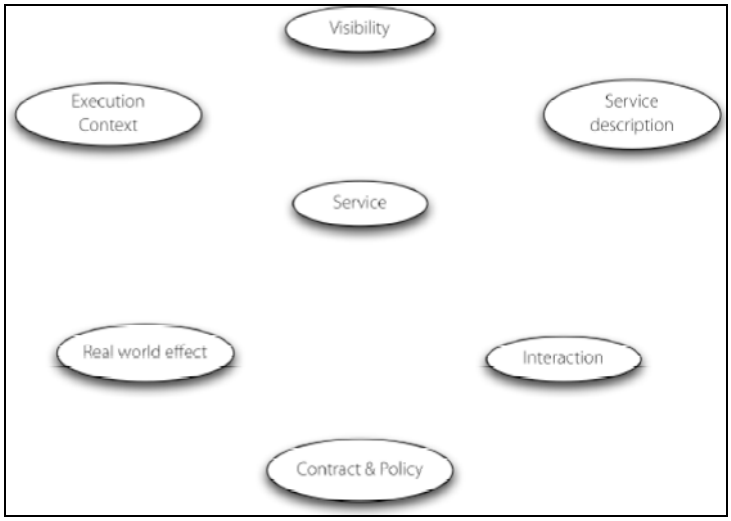



**Figure 101 Principal concepts of the OASIS SOA Reference Model**

The following description is then explaining what I have earlier described as "SOA heartbeat".

**Comments:**

The reference model shows an abstract framework for understanding the relationships between the different mentioned concepts. The method is very formal and explains the relationships around the service concept. As a reference model does not indicate any method nor any types of models used etc., the content can just be used to verify and validate some concepts such as “service description” or “interaction”.

The following list is resuming the issues described in the SOA Domain model structured in SOA Phases to allow better readability. Second, the table is showing which of the analysed SOA methods is covering the presented issues and to what degree the issues are covered. The result is identical to the summary model of method analysis, but just presented in another format.